ISSN 0860-1941

CENA 220 zł

mikroklan

vademecum techniki mikrokomputerowej

ZESZYT 2

Mikrokomputerowe abc

Zastosowanie w sekretariacie

Systemy operacyjne UNIX

Standard Centronics

Wszystko o minidyskietkach

TURBOTAPE 64

Superlife

ATARI 800 XL

WYDAWNICTWO NOT SIGMA

Przyszłość sekretariatu

mikroklan ZESZYT 2

Wprowadzenie
w zagadnienia
techniki
mikrokomputerowej

Praca zbiorowa

Opracował zespół redakcji
miesięcznika INFORMATYKA

Teresa Jabłońska, Władysław Klepacz,
Krzysztof Kontek, Ireneusz Myzik,
Jerzy Orkiszewski, Andrzej J. Piotrowski
Adam Pluta, Zbigniew Pojmański,
Mateusz Stryjecki (oprac. graficzne)

WYDAWNICTWO CZASOPISM
I KSIĄŻEK TECHNICZNYCH

SIGMA

PRZEDSIĘBIORSTWO NACZELNEJ
ORGANIZACJI TECHNICZNEJ

00-950 Warszawa skrytka 1004
ul. Biała 4

ISSN 0860-1941
Wydawnictwo Czasopism
i Książek Technicznych
NOT-SIGMA
Warszawa 1986

Drukowano na zlecenie ARS POLONA

Redakcja: 01-517 Warszawa,
ul. Mickiewicza 18 m. 17, tel. 39 11 34
Skład: technika fotoskładu Eurocat 150
Wydawnictwo NOT-Sigma
Druk: Bohmann Druck und Verlag
GmbH&Co. KG, Wiedeń. Austria
Nakład 100 000 egz. P-47
Cena 220 zł

SPIS TREŚCI

SPRZĘT

- Pamięci na cenzurowanym
  Jerzy Orkiszewski 6
- Standard Centronics
  Andrzej J. Piotrowski 21
- Pamięć zewnętrzna mikrokomputerów na minidyskach elastycznych (1)
  Adam Pluta 27

OPROGRAMOWANIE

- Superlife – symulacja rozwoju żywych komórek
  Jerzy Szafranek 11
- TURBOTAPE 64
  Klaudiusz Dybowski, Michał Silski 13
- Czym jest UNIX (2). Polecenia systemowe
  Janusz Zalewski 18
- UNIX w drodze po sukces
  micro 20
- Ukryte gwiazdy
  micro 22
- Symulacja drgań w dwóch wymiarach
  Bernard Piotr Żurawski 29

ZASTOSOWANIA

- Luksus czy pożyteczne narzędzie?
  micro 8

INFORMATYCZNE ABC

- BASIC dla początkujących (2)
  micro 2
- Bit, Bajt i Spółka
  micro 4

EDUKACJA

- Oczami nauczycieli
  Krzysztof Kontek 25

OPIS

- ATARI 800 XL
  Jerzy Orkiszewski 15

KORESPONDENCJA

- Jak pisać do MIKROKLANU 32

NOWOŚCI 10,31

CENY 32

BASIC dla początkujących

W poprzednim odcinku kursu języka BASIC była mowa o tym, że praca komputera składa się z trzech etapów: wprowadzania danych, przetwarzania i wyprowadzenia rezultatów.

Przypomnijmy sobie ćwiczenie z mnożenia. Zapiszemy je teraz w postaci programu, który zobrazuje normalny sposób pracy komputera. Wystukajmy na klawiaturze:

```
10 INPUT A
20 A=A*A
30 PRINT A
```

i rozpocznijmy realizację programu wystukując słowo **RUN.** Na ekranie pojawi się znak zapytania. Trzeba teraz wprowadzić jakąś liczbę i wcisnąć klawisz *RETURN,* a w następnym wierszu na ekranie pojawi się wynik. W ten sposób mikrokomputer wykonał program napisany w języku BASIC.

Nasz program składa się z trzech linii zaczynających się liczbą. Każda liczba określa numer linii. Za numerami linii napisane są instrukcje. Wykonywane są one w kolejności wzrastających numerów. Najpierw więc wykonana zostanie instrukcja w linii *10,* potem w *20,* a na koniec *30.*

Linia *10* realizuje wprowadzenie danej – jednej liczby. Służy do tego instrukcja INPUT. Gdy komputer wykona tę instrukcję na ekranie pojawi się znak zapytania zachęcający do wprowadzenia liczby. Wystukana na klawiaturze liczba po wciśnięciu klawisza *RETURN* zostanie zapamiętana w pewnej komórce pamięci, którą oznaczyliśmy literą *A.* Do tej komórki odwołujemy się jeszcze w liniach *20* i *30.* Ponieważ można zapisywać liczby o różnych wartościach, zmieniających się w trakcie realizacji programu, można powiedzieć, że w ten sposób realizowane jest zapamiętanie wartosci przypisanej zmiennej *A.*

Instrukcja w linii *20* realizuje przetwarzanie. Z pamięci pobierana jest wartość przypisana zmiennej *A* i mnożona przez samą siebie. Obliczona wartość zostaje następnie zapisana w komórce pamięci oznaczonej przez *A.* Po wykonaniu instrukcji z linii *20* następuje więc zmiana wartości przypisanej zmiennej *A.* Jej nową wartością jest wartość z linii *10* podniesiona do kwadratu. Znak równości w języku BASIC jest czymś innym niż w arytmetyce. Oznacza on przypisanie zmiennej, występującej po lewej stronie znaku równości, wartości wyrażenia występującego po jego prawej stronie.

Instrukcja w linii *30* wyprowadza rezultat – wynik obliczeń. Jest to omawiana już w pierwszej części instrukcja *PRINT* – tyle, że zamiast liczby pojawia się w niej nazwa zmiennej. Instrukcja ta powoduje w tym wypadku pobranie z pamięci aktualnej wartości przypisanej zmiennej *A* i wyprowadzenie jej na ekran. Wartość ta będzie nadal istniała w pamięci i może być powtórnie wyświetlona na ekranie lub użyta do dalszych obliczeń.

POLECENIE A INSTRUKCJA

Polecenie podawane w trybie bezpośrednim, nie jest zapamiętywane na stałe, lecz wykonywane natychmiast po wciśnięciu klawisza *RETURN.*

To samo polecenie wprowadzone w trybie edycji, jako ponumerowana linia tworząca program, nazywa się instrukcją. Tak więc zapis:

```
RUN
```

jest poleceniem, natomiast zapis:

```
80 RUN
```

linią programu z instrukcją *RUN.*

Program jest przechowywany w pamięci komputera. Może być w dowolnej chwili wyświetlony po wydaniu polecenia *LIST.* Instrukcje składające się na program są realizowane po podaniu polecenia *RUN* i wciśnięciu klawisza *RETURN.*

INSTRUKCJA INPUT

Operowanie samymi tylko liczbami jest niewygodne i łatwo prowadzi do pomyłek. Zazwyczaj chcemy dokładnie wiedzieć, co należy w danej chwili wprowadzić do komputera. W tym celu możemy w instrukcji *INPUT* umieścić tekst określający, jakie dane należy wprowadzić. Wpiszmy z klawiatury:

```
10 INPUT "WARTOSC"; A
```

Wpisanie linii z tym samym numerem, co już istniejąca, spowoduje wymazanie z pamięci tej linii i wstawienie na jej miejsce nowej. Teraz nasz program będzie wyglądał następująco:

```
10 INPUT "WARTOSC"; A
20 A=A*A
30 PRINT A
```

Można się o tym przekonać, podając polecenie LIST, które spowoduje wyświetlenie aktualnej wersji programu. Za słowem *INPUT,* pomiędzy znakami cudzysłowu, może znajdować się dowolny tekst, który zazwyczaj opisuje wprowadzaną zmianę. Musi on być obowiązkowo oddzielony od nazwy zmiennej średnikiem.

Po rozpoczęciu realizacji naszego programu na ekranie pojawi się tekst z instrukcji *INPUT,* zaś migający kursor przyjmie pozycję bezpośrednio za wyświetlonym tekstem (w tym przypadku tekstem: *WARTOSC*). Gdy teraz wprowadzimy dowolną liczbę, na ekranie pojawi się (bez komentarza) kwadrat tej liczby jako wynik działania programu.

ZMIENNE

Teraz nieco o zmiennych i stałych. Nazwy zmiennych rzeczywistych w mikrokomputerze Commodore 64 mogą być jedno- lub dwuznakowe. Na pierwszym miejscu występuje zawsze litera, na drugim litera lub cyfra. Prawidłowymi nazwami są więc: *A, B,..., Z, AA, AB, ..., AZ, BA, ..., ZZ, A0, A1, ..., Z8, Z9.* Po włączeniu komputera wszystkie zmienne mają wartość 0. W trakcie realizacji programu wartości przypisane zmiennym mogą ulegać zmianie.

Zmienne oznaczone zgodnie z powyższymi regułami są typu rzeczywistego, tzn. ich wartości są liczbami rzeczywistymi z przecinkiem dziesiętnym. Zmienna rzeczywista zajmuje w pamięci 5 elementarnych komórek, czyli bajtów.

Drugim typem zmiennych są zmienne całkowite. Dla wprowadzenia rozróżnienia, ich nazwy zakończone są znakiem procentu (np. *A%, AB%, A1%*). Liczby przypisywane zmiennym całkowitym nie mają części ułamkowej, zaś w pamięci zajmują tylko 2 bajty.

Trzecim typem są zmienne łańcuchowe (ang. string – łańcuch). Zamiast znaku % mają na końcu nazwy znak dolara *$.* Ich wartościami nie są liczby, lecz łańcuchy znaków mogące tworzyć teksty. Na zmiennych tych nie można wykonywać operacji arytmetycznych. Istnieją jednak specjalne instrukcje, które omówimy później, służące do przekształcania łańcuchów znaków i do wyprowadzania ich na ekran.

Można używać jednocześnie wszystkich typów zmiennych. Tak więc np. *A, A%* i *A$* są oznaczeniami trzech różnych zmiennych: rzeczywistej, całkowitej oraz łańcuchowej i mogą występować w tym samym programie.

Zmienne można uważać za miejsca w pamięci, w których przechowywane są przypisane im wartości liczbowe albo łańcuchy znaków (teksty).

STAŁE

W przeciwieństwie do zmiennych, stałe nie mogą zmieniać swej wartości. Spotkaliśmy się już z nimi w pierwszej części, przy pracy w trybie bezpośrednim. Liczby wprowadzane po instrukcji *PRINT,* używanej tam w skróconej formie – w postaci znaku zapytania, były właśnie stałymi. W tamtym wypadku były to liczby całkowite. Poznaliśmy też już stałą łańcuchową. Był to ciąg znaków *"WARTOSC"* w instrukcji *INPUT* z naszego programu. Stałe łańcuchowe muszą być zawsze ujęte z obu stron w cudzysłów.

Stałymi liczbowymi (całkowitymi i rzeczywistymi) są bezpośrednio podane w programie wartości liczbowe, zaś stałymi łańcuchowymi są łańcuchy znaków ujęte w cudzysłów, np. *"TEKST".*

Tekst wyprowadzany przez instrukcję INPUT może być tylko stałą łańcuchową, nigdy zaś zmienną.

INSTRUKCJA PRINT

Przedstawimy teraz kolejną wersję instrukcji *PRINT.* Wystukajmy na klawiaturze:

```
30 PRINT "KWADRAT LICZBY JEST ROWNY", A
```

Za pomocą jednej instrukcji *PRINT* można wyprowadzić na ekran zmienne oraz stałe, zarówno liczbowe, jak i łańcuchowe. Poszczególne wielkości występujące w instrukcji *PRINT* należy oddzielić od siebie przecinkami.

Przecinek, podobnie jak średnik, ze względu na funkcję, jaką spełnia w zapisie instrukcji, nazywany jest separatorem.

Rozpocznijmy realizację programu poleceniem *RUN*. Gdy komputer wyświetli słowo *WARTOSC,* wprowadźmy liczbę *3* (należy pamiętać o klawiszu *RETURN!*). Na ekranie natychmiast pojawi się napis:

KWADRAT LICZBY JEST ROWNY 9.

Wprowadźmy teraz następujące wiersze:

```
25 B=A*A
40 PRINT "CZWARTA POTEGA JEST ROWNA", B
```

Podajmy polecenie *LIST*. Na ekranie wyświetli się nasz program. Będzie on teraz wyglądał następująco:

```
10 INPUT "WARTOSC"; A
20 A=A*A
25 B=A*A
30 PRINT "KWADRAT LICZBY JEST ROWNY", A
40 PRINT "CZWARTA POTEGA JEST ROWNA", B
```

Działanie programu jest następujące:
W linii *10* zostaje wprowadzona wartość zmiennej *A*, w linii *20* wartość ta jest podnoszona do kwadratu, w linii *25* zmienna *B* przyjmuje podniesioną do kwadratu wartość zmiennej *A*, czyli jest czwartą potęgą liczby wprowadzonej w linii *10*.

Rozpocznijmy teraz realizację programu, podając polecenie *RUN*. Po słowie *WARTOSC* wprowadzimy liczbę np. *3*. Po zakończeniu wykonania programu na ekranie zostaną wyświetlone napisy:

```
WARTOSC 3
KWADRAT LICZBY JEST ROWNY 9
CZWARTA POTEGA JEST ROWNA 81
```

Proponujemy teraz Czytelnikom samodzielne opracowanie programu, który wczytuje liczbę, a następnie wyświetla jej pierwiastek kwadratowy oraz drugą i trzecią potęgę. Jeszcze jedna rada: należy używać nazw mnemotechnicznych, wiążących zmienną z jej znaczeniem.

W zaproponowanym programie można użyć np. następujących oznaczeń:
L – wprowadzana liczba,
P – pierwiastek kwadratowy,
K – kwadrat liczby,
S – sześcian liczby.

Przykładowo program może wyglądać następująco:

```
10 INPUT "WPROWADZANA LICZBA"; L
20 P=SQR (L)
30 K=L*L
40 S=L*L*L
50 PRINT "PIERWIASTEK", P
60 PRINT "KWADRAT",K
70 PRINT "SZESCIAN",S
```

Po wydaniu polecenia realizacji programu na ekranie pojawiają się napisy i wyniki liczbowe. Są one jednak rozmieszczone w mało przejrzysty sposób. Poszczególne liczby wypisywane są w różnych kolumnach. Nauczmy się więc organizować wyniki w czytelny sposób.

PRZECINEK JAKO TABULATOR

Jak zapisać nasz program, aby wyniki zostały wyprowadzone w formie tabeli? Są dwie możliwości. Pierwsza polega na dodaniu do krótszych łańcuchów (u nas: KWADRAT i SZESCIAN) znaków odstępu, czyli spacji, tak by ich długość, tzn. liczba znaków, była zawsze taka sama jak najdłuższego łancucha (u nas *PIERWIASTEK*). ***W stałych łańcuchowych mogą występować znaki spacji.*** Drugi sposób polega na napisaniu w liniach 50, 60 i 70 nie po jednym przecinku, lecz po dwa. Dodatkowy przecinek spowoduje skok do najbliższej pozycji tabulacyjnej. Przecinek może więc w programie spełniać podwójną rolę: separatora i tabulatora.

W pierwszym przypadku nasz program wygląda tak:

```
10 INPUT "WPROWADZANA LICZBA";L
20 P=SQR (L)
30 K=L*L
40 S=L*L*L
50 PRINT "PIERWIASTEK",P
60 PRINT "KWADRAT",K
70 PRINT "SZESCIAN",S
```

A w drugim przypadku tak:

```
10 INPUT "WPROWADZANA LICZBA";L
20 P=SQR (L)
30 K=L*L
40 S=L*L*L
50 PRINT "PIERWIASTEK" ,,P
60 PRINT "KWADRAT" ,,K
70 PRINT "SZESCIAN" ,,S
```

Programy służące do rozwiązania tylko jednego konkretnego problemu rachunkowego, jak na przykład program obliczający pierwiastek, kwadrat i sześcian liczby *4*, nie mają większego sensu. Program opłaca się pisać wtedy, gdy może być on realizowany dla różnych danych. Zatem zamiast stałych liczbowych powinny występować zmienne, dla których wartości wprowadza się z klawiatury.

INSTRUKCJA RUN

RUN znaczy po angielsku biegnij. Słowo *RUN* może być wprowadzone bez numeru wiersza – wtedy zostanie potraktowane jako polecenie i natychmiast rozpocznie się wykonywanie znajdującego się w komputerze programu. Może też być wprowadzone z numerem wiersza. Wprowadźmy:

```
80 RUN
```

Do naszego programu została więc dopisana na końcu jeszcze jedna linia. Gdy program dojdzie do tej linii, zostanie uruchomiony na nowo. Wartościami początkowymi zmiennych w kolejnym przebiegu będą wartości końcowe zmiennych z poprzedniego przebiegu. Jeżeli po linii z instrukcją RUN zostaną dopisane inne linie, to będą one pomijane.

DODATEK DLA UŻYTKOWNIKÓW ZX SPECTRUM

Autorzy języka BASIC zaproponowali dla instrukcji przypisania słowo *LET*. Przykładowo może ona mieć postać:

```
LET A=1
```

Zapis taki oznacza, że od tego miejsca zmiennej *A* przypisuje się w realizowanym programie wartość jeden. Po prawej stronie znaku = może występować dowolna funkcja lub wyrażenie arytmetyczne, po obliczeniu którego dostaniemy konkretną liczbę. W wyrażeniu mogą występować dowolne zmienne, których bieżąca wartość jest wykorzystywana do wyznaczenia wartości wyrażenia. Nic nie stoi na przeszkodzie, by wykorzystywać również wartość przypisaną zmiennej występującej po lewej stronie znaku = , np.:

```
LET A=A*A
```

Wiele interpreterów języka BASIC dopuszcza możliwość pominięcia *LET*, choć dla osób przyzwyczajonych do konwencjonalnego znaczenia znaku = zapis *A=A*A* nie ma sensu. Natomiast interpreter języka BASIC w ZX Spectrum nie dopuszcza możliwości pominięcia słowa kluczowego *LET*.

Przy stosowaniu instrukcji *INPUT* w ZX Spectrum zamiast znaku ? pojawia się na ekranie kursor *L*. Jest on wyświetlany u dołu ekranu.

ZX Spectrum pozostawia większą swobodę w wybieraniu nazw dla zmiennych. Nazwą zmiennej rzeczywistej może być dowolnie długi ciąg liter i cyfr pod warunkiem, że zaczyna się on od litery. W nazwie zmiennej mogą również występować spacje (odstępy), lecz niedopuszczalne są inne znaki (np. * –,$, itp.).

W ZX Spectrum nie zastosowano skróconej wersji zapisu dla zmiennych całkowitych. Wszystkie zmienne liczbowe są przechowywane w pamięci w tzw. postaci zmiennoprzecinkowej. Jeden bajt pamięci jest przeznaczony na przechowywanie wykładnika, a cztery bajty na mantysę (ze znakiem). Niezależnie od wartości, przechowywana w pamięci liczba zajmuje pięć bajtów.

ZX Spectrum dopuszcza stosowanie zmiennych łańcuchowych. Ich nazwy składają się z pojedynczych liter, po których podawany jest symbol dolara *$*. Ponieważ Spectrum nie rozróżnia w nazwach zmiennych małych i dużych liter (również dla zmiennych rzeczywistych) w programie może wystąpić tylko 26 różnych zmiennych łańcuchowych.

Instrukcja PRINT w Spectrum dopuszcza stosowanie trzech separatorów: , ; oraz '. Przecinek może być podobnie jak w C64 stosowany jako tabulator – następujące po nim wyrażenie jest wyświetlane od połowy ekranu. Jeżeli wcześniej wyświetlane znaki przeszły już za połowę ekranu, to nastąpi przejście do następnego wiersza. Średnik powoduje, że dalsza część napisu jest wyświetlana bezpośrednio za dotychczas wyprowadzonymi znakami. Jeżeli napiszemy:

```
PRINT "W domu jest";10;"kotów"
```

na ekranie pojawi się napis:

W domu jest10kotów

Nie należy więc zapominać o wprowadzeniu dodatkowych spacji (po *jest* i przed *kotów*). Wykorzystanie jako separatora pojedynczego cudzysłowu wymusza przejście do nowej linii przed wyświetleniem dalszej części napisu.

Istnieją jeszcze dwie możliwości organizowania postaci wyprowadzanych napisów. Pisząc:

PRINT "napis 1";TAB n;"napis 2"

spowodujemy, że tekst *napis 2* rozpocznie się od *n-tej* kolumny. Możliwe jest też napisanie:

PRINT "napis 1";AT w,k;"napis 2"

Tekst *napis 2* zostanie wyświetlony w wierszu o numerze w od *k-tej* kolumny.

Trochę inaczej wygląda zagadnienie dla instrukcji *INPUT*. Ekran Spectrum jest podzielony na dwie części. Dolna część ekranu (wiersze numer 22 i 23) jest przeznaczona do wyświetlania komunikatów i właśnie wprowadzania informacji instrukcją *INPUT*. Granica między obu częściami ekranu jest płynna, tzn. dolny obszar rozszerza się w miarę potrzeby, zajmując ewentualnie więcej wierszy. Separatory wykorzystywane w instrukcji *INPUT* spełniają taką samą funkcję jak w instrukcji *PRINT*. Można również wykorzystywać *TAB* i *AT*. Wprowadzając:

INPUT AT 0,0;"napis 1";a;AT 1,0;"napis 2";b

spowodujemy jednak, że teksty będą pojawiać się nie u góry, lecz od dołu ekranu – w razie konieczności przesuwając się w górę. Przesuwający się w górę tekst z instrukcji *INPUT* może od pewnego momentu kolidować z napisami wykonanymi instrukcją *PRINT*. W takiej sytuacji całość ekranu jest przesuwana w górę (najwyższy wiersz wypada z ekranu). Warto tu zwrócić uwagę, że normalnie tekstu wypisanego instrukcją *PRINT* nie można dopełnić tekstem podanym w instrukcji *INPUT*.

W Spectrum, podobnie jak w C64, można podać instrukcję *RUN* w programie. Wznawia ona realizację programu z jednoczesnym wyczyszczeniem ekranu i wyzerowaniem wszystkich zmiennych. Jeżeli w instrukcji *RUN* podamy jako argument numer linii, realizacja programu rozpocznie się właśnie od tej linii.

SŁOWNICZEK TERMINÓW OBCOJĘZYCZNYCH

AT *(et)* – wskazanie miejsca: na, w, itp.
INPUT *(input)* – wprowadz
LET *(let)* – niech... (instrukcja przypisania wartości)
LIST = listing *(listing)* – listowanie (drukowanie treści programu)
PRINT *(print)* – drukuj; wyświetl na ekranie
RUN *(ran)* – biegnij (rozpoczęcie realizacji programu)
SQR = square root *(skłer ruut)* – pierwiastek kwadratowy
TAB = tabulation *(tabjulejszyn)* – tabulacja

AJP

Bit, Bajt i Spółka

Dzięki rozwojowi myśli technicznej i postępowi technicznemu hasło „komputer dla każdego" przestało być mrzonką. Jeszcze przed kilku laty tylko duże firmy lub milionerzy mogli sobie pozwolić na komputer, który pomagałby im w prowadzeniu interesów. Przeciętni obywatele musieli się zadowolić kalkulatorami kieszonkowymi, wykonującymi niewiele więcej niż cztery podstawowe działania arytmetyczne. Dzisiaj sytuacja zmieniła się radykalnie. Mikrokomputer w wielu krajach często kosztuje mniej niż telewizor.

Na rynku pojawiło się wiele typów mikrokomputerów, a więc przed zakupem należy się zastanowić: czy mikrokomputer droższy rzeczywiście będzie dla nas lepszy? A może wystarczy nam tańszy model? Czy potrzebny nam od razu samochód, skoro wystarczy rower? Aby odpowiedzieć na te pytania trzeba zapoznać się przynajmniej z ogólnymi zasadami budowy i działania komputera. Nie jest to wcale takie skomplikowane, o czym przekonacie się po przeczytaniu tego artykułu. Da on Wam podstawową wiedzę, niezbędną przy podejmowaniu decyzji o wyborze mikrokomputera.

PAMIĘĆ I JEJ POJEMNOŚĆ

Co to znaczy, że komputer ma dużą pamięć (dużą pojemność pamięci)? Spróbujcie wyobrazić sobie pamięć jako magazyn, zbiór elektronicznych komórek wypełnionych danymi i rozkazami dla komputera. Komórki te, w żargonie używanym przez fachowców nazywa się czasem lokacjami (ang. locations).

Każda z komórek ma własne oznaczenie, czyli swój adres. Pod każdym adresem zapamiętana jest podstawowa jednostka informacji, czyli bajt (ang. byte). Bajt składa się z ośmiu mniejszych jednostek – bitów. Bit (ang. BInary digiT – cyfra dwójkowa) jest najmniejszą, niepodzielną jednostką informacji. Może on przybierać jedną z dwóch wartości, umownie oznaczanych przez 0 i 1. Bajt złożony z ośmiu dwuwartościowych bitów, może mieć jedną z 256 wartości (2^8). Wartości te mogą symbolizować znaki alfanumeryczne (cyfry, litery i znaki specjalne), dane graficzne lub poszczególne rozkazy mikrokomputera.

Na ogół największym adresem w popularnym mikrokomputerze jest 65 535. Naturalnie są też mikrokomputery bardziej skomplikowane, o większej przestrzeni adresowej.

ROM I RAM

Aby uniknąć konieczności wielokrotnego wprowadzania do pamięci tych samych podstawowych operacji i danych, opracowano specjalny rodzaj pamięci, zwany ROM (ang. Read Only Memory – pamięć tylko do odczytu), a po polsku *pamięcią stałą*. Zawartość tej pamięci jest wprowadzana przez wytwórcę komputera i nie znika po wyłączeniu komputera z sieci zasilającej. Komputer może odczytywać zawartość ROM i wykonywać znajdujące się w niej instrukcje, lecz nie może w niej niczego zmienić. Pamięć stała zajmuje tylko część przestrzeni adresowej.

Pozostała część przestrzeni adresowej przeznaczona jest na pamięć, w której umieszczone są dane i instrukcje wprowadzane przez użytkownika. Informacje te nie są na stałe umieszczone w pamięci, lecz znikają po wyciągnięciu z gniazdka wtyczki kabla zasilającego mikrokomputer. Pamięć elektroniczna, w której możemy zapisywać informacje, nazywana jest pamięcią RAM. Jest to niezbyt poprawne (skrót pochodzi od angielskich słów Random Access Memory – pamięć o dostępie swobodnym), gdyż zarówno ROM jak i RAM są pamięciami o dostępie swobodnym. Oznacza to, że można bezpośrednio odwoływać się do każdej lokacji, podając jej adres. To, co istotnie różni oba rodzaje pamięci – to nieulotność informacji w ROM i ulotność w RAM.

ISTOTNA JEST ZAWARTOŚĆ

Niektórzy producenci zachwalają swe komputery podkreślając, że zawierają one więcej komórek pamięci stałej (ROM) niż wyroby konkurentów. Spokojnie możecie zignorować ich argumentację. Przecież istotna jest nie liczba komórek tej pamięci, lecz liczba i jakość funkcji wprowadzanych do niej na stałe. Najlepszym rozwiązaniem jest możliwie duża liczba funkcji, zapamiętana w możliwie małej liczbie komórek, tzn. program dobrze upakowany. Takie programy są jednak rzadkością. Znaczna liczba producentów dodając dodatkowe funkcje po prostu rozszerza obszar ROM.

Zupełnie odwrotnie ma się sprawa z RAM. Jej pojemność określa maksymalną długość Waszych programów, a także szybkość, z jaką działa komputer. Im więcej danych można na raz przepisać z pamięci zewnętrznej (np. taśmy magnetofonowej lub dyskietki) do RAM, tym rzadziej trzeba tę czynność powtarzać. Komputer nie traci czasu na przesyłanie danych i może się zająć ich przetwarzaniem.

Musicie się też orientować, jaką objętość mają dostępne w sprzedaży programy, z których zamierzacie korzystać. Większość z nich potrzebuje obszaru 8 lub 16 KB (1 KB = 1 kilobajt = 1024 bajty). Zdarzają się także programy o długości 32 KB i więcej. A zatem, jeszcze przed zakupem trzeba wiedzieć, do czego będzie stosowany Wasz komputer. Jeśli pojemność RAM będzie większa niż wymaga tego najdłuższy program, wszystko będzie w porządku. Jeśli natomiast będzie za mała, program nie da się uruchomić i nastąpi tzw. załamanie systemu (ang. crash).

Wybór komputera powinien być poprzedzony starannym rozpatrzeniem potrzeb i

kosztów: ile możecie lub chcecie zainwestować na początek? Jak duża jest niezbędna pojemność pamięci?

Należy też wziąć pod uwagę fakt, że ceny RAM wciąż spadają. Dwa lata temu warto było się zastanawiać nad wyborem między 8 a 16 KB. Obecnie prawdopodobnie nie warto kupować mniej niż 64 KB. Na ogół pojemność pamięci wynosi 16, 32, 48, 64 lub 128 KB. Nowsze modele mogą mieć pojemność 256 KB lub większą. IBM-PC oferowany jest często z RAM o pojemności 720 KB. Ceny są jednak oczywiście odpowiednio większe.

URZĄDZENIA ZEWNĘTRZNE (wejścia-wyjścia)

Jeżeli mikrokomputer ma być czymś więcej niż tylko kiepskim i zbyt kosztownym urządzeniem nagrzewającym pomieszczenie, powinien być wyposażony w urządzenia wejścia-wyjścia. Tylko dzięki nim można kontaktować się z maszyną.

Mówiąc „komputer", mamy na ogół na myśli obudowę z tworzywa sztucznego lub metalu z klawiaturą i przyłączony do niej monitor ekranowy lub telewizor. W rzeczywistości jest to system komputerowy z urządzeniem wejściowym w postaci klawiatury, jednostką centralną i urządzeniem wyjściowym w postaci telewizora lub monitora.

Do pracy z komputerem niezbędne są co najmniej klawiatura i telewizor. Można ogólnie powiedzieć, że im więcej jest urządzeń zewnętrznych, tym komputer jest bardziej wszechstronny. W komputerach domowych i osobistych klawiatura jest prawie zawsze scalona z ich obudową. Odpowiedni telewizor na ogół został już wcześniej kupiony. To wystarcza na początek; później w razie potrzeby można wzbogacić system dołączając drukarkę lub pamięć zewnętrzną.

EKRAN

Ekran jest jedną z najważniejszych części komputera. To właśnie na ekranie obserwujecie wyniki działania, przedstawione w odpowiedniej formie. Jeśli komputer dysponuje barwną grafiką, warto mieć kolorowy telewizor lub monitor, aby w pełni wykorzystać jego możliwości. W przeciwnym wypadku wystarczy telewizor czarno-biały. Koniecznie sprawdźcie zawsze jakość obrazu. Nie zapominajcie, że podczas pracy z komputerem wciąż macie – w dosłownym znaczeniu tych słów – ekran przed oczami.

Wiele modeli ma już wbudowany monitor, do innych podłącza się własny telewizor.

Format obrazu zależy od rodzaju komputera; decydująca jest tu liczba wierszy (w pionie) i kolumn (w poziomie) znaków. Najpopularniejsze formaty to 32, 40, 64 i 80 kolumn oraz 1 (komputery kieszonkowe) 16, 24 i 25 wierszy. Im większy format, tym więcej informacji można jednocześnie wyświetlić. Weźmy dla przykładu format o 32 kolumnach i 16 wierszach – oznacza to, że jednocześnie można wyświetlić 512 znaków (32 × 16).

Oczywiście także i tutaj należy rozważyć koszty i potrzeby: im większy format, tym – jakże mogłoby być inaczej – większa cena.

KLAWIATURA

Przy poszukiwaniu odpowiedniej klawiatury spotkaliście się zapewne z określeniem *human engineering*, co można przetłumaczyć jako *projektowanie zorientowane na człowieka*. Jest to ostatnio modne pojęcie, spotykane także przy nowych opracowaniach z innych dziedzin techniki. Oznacza ono, że przy projektowaniu wyrobu uwzględnia się nie tylko wszystkie wymagania techniczne stawiane przed wyrobem, ale także i cechy, które zapewniają użytkownikowi komfort w jego stosowaniu. Na przykład, w wielu klawiaturach klawisze z cyframi i symbolami operacji zgrupowane są po prawej stronie, w postaci dodatkowej klawiatury podobnej do stosowanej w kalkulatorach. Ułatwia to pracę tym użytkownikom, którzy często przeprowadzają operacje matematyczne. Całość klawiatury ma wtedy nieco większą powierzchnię, lecz nie można tego uważać za wadę.

W żadnym przypadku nie należy rezygnować z próbnego wystukania na klawiaturze danych i instrukcji, które zamierzacie często stosować. W ten sposób najwcześniej rozstrzygniecie, czy rozłożenie specjalnych klawiszy funkcyjnych jest odpowiednie. Jeśli np. klawisz wymazujący ostatnio wprowadzony znak został tak usytuowany, że wciąż następuje jego mimowolne wciśnięcie, to odkrycie tego błędu konstrukcyjnego przed zakupem, uchroni Was przed późniejszymi kłopotami.

W użyciu są dwa rodzaje klawiatur: płaskie klawiatury sensorowe, reagujące na najlżejsze dotknięcie oraz zwykłe klawiatury mechaniczne, spotykane w maszynach do pisania.

Pod względem funkcjonalnym klawiatury sensorowe niczym się nie różnią od tradycyjnych. Zbudowane są z folii plastikowej, na którą nadrukowano pola klawiszy. Folia ta, choć cienka, jest bardzo wytrzymała oraz – co czyni jej stosowanie szczególnie atrakcyjnym – dość tania.

Klawiatura taka jest oczywiście łatwiejsza do czyszczenia i dzięki temu szczególnie odpowiednia do stosowania w „niesprzyjających warunkach użytkowania". Kurz i wilgoć przestają być problemem. Nie trzeba się już martwić palcami w czekoladzie ani przyklejoną gumą do żucia – cóż za pomyślna wiadomość dla rodziców wieku elektroniki!

Mimo tych zalet klawiatury sensorowe stosowane są tylko w najtańszych modelach. Powód jest następujący: przy stosowaniu płaskiej klawiatury niemożliwe jest odnalezienie, za pomocą samego dotyku, klawiszy. Bowiem są one tylko namalowane. To tak, jakby szukać na kartce papieru miejsca do naciśnięcia. Nie ma żadnej możliwości stwierdzenia za pomocą zmysłu dotyku, czy dany klawisz został faktycznie wciśnięty. Powinniście oczywiście wypróbować także takie rozwiązanie. Może się okazać, że jego niedostatki nie okażą się dla Was znaczące w zestawieniu z niską ceną.

Wszystkie zalety klawiatury sensorowej w przypadku klawiatury mechanicznej zamieniają się w wady: jest ona droga, wrażliwa na uszkodzenia i trudna do czyszczenia.

Używane są tu przełączniki stykowe, które łatwo się utleniają i brudzą. Kurz, płyny i wilgoć są – w najprawdziwszym tych słów znaczeniu – śmiertelnymi wrogami. Klawisze mechaniczne wymagają więc napraw.

Wszystko to nie jest w stanie przekreślić podstawowej zalety: nie trzeba przez cały czas spoglądać na klawiaturę. Dzięki oporowi, jaki dany klawisz stawia palcom, można stwierdzić, czy został on wciśnięty. Opór klawiszy, ich wielkość i głębokość wcisku są różne, podobnie jak i wielkość całej klawiatury: od klawiatury komputera formatu kieszonkowego, aż do standardowej klawiatury maszyny do pisania.

Droższe klawiatury stawiają lekki, ale wyczuwalny opór, a o prawidłowym włączeniu klawisza świadczy cichy dźwięk wydawany przez kontaktron. Tańsze modele można rozpoznać po braku tego odgłosu. Ocena klawiatury jest jednak sprawą wysoce subiektywną. W celu dokonania najtrafniejszego wyboru należy siąść i wypróbować ją w działaniu.

Sprawdźcie też, czy klawiatura może zapewnić odczytywanie wszystkich znaków również przy bardzo szybkim wciskaniu klawiszy. Własność ta jest niezbędna, jeśli nie chcecie po wpisaniu danych poszukiwać zgubionych znaków.

Niepożądany jest także jeszcze jeden efekt, związany z drganiem zestyków klawiszy. Drgania te trwają po wciśnięciu klawisza zaledwie ułamek sekundy – jest to jednak wystarczająco długo, aby zostały one odczytane jako kilka kolejnych wciśnięć. Zamiast jednego znaku komputer odbiera wtedy kilka takich samych. Efekt ten eliminuje się stosując specjalne układy lub metody programowe. Nie w każdym modelu metoda ta jest jednak całkowicie skuteczna.

Najlepiej jest mieć znajomych, którzy pracują na klawiaturze podobnej lub identycznej z tą, którą chcecie nabyć. W każdym przypadku bezpieczniej skorzystać z ich opinii niż wierzyć zapewnieniom producentów czy sprzedawców.

Przy wyborze możecie się kierować następującym kryterium: jeśli sprzęt przeznaczony jest dla Was, wybierzcie klawiaturę mechaniczną, jeśli dla dzieci – sensorową. Jeśli często będziecie używali operacji matematycznych, wybierzcie model z wydzielonymi klawiszami cyfr.

Poprzez klawiaturę komunikujecie się z komputerem – nie wolno więc nie doceniać jej znaczenia.

GRAFIKA I ZESTAW ZNAKÓW

Grafikę stanowią tworzone przez komputer ilustracje i wykresy. Mówiąc ogólnie, każdą informację nietekstową i nienumeryczną możemy uważać za grafikę. Spotyka się komputery z lepszymi i gorszymi możliwościami graficznymi. Modele prostsze mają na ogół zestaw standardowych znaków graficznych przyporządkowanych oddzielnym klawiszom. Znaki te są wyświetlane lub drukowane, podobnie jak pozostałe znaki alfanumeryczne.

W modelu z dużą rozdzielczością każdy oddzielny punkt na ekranie może być niezależnie sterowany i wyświetlony. Otwiera to możliwości generowania niezwykle dokładnych i szczegółowych obrazów. Ekran stanowi matrycę tysięcy małych punktów świetlnych (ang. pixel – skrót od *picture element* – element obrazu), całkowicie od siebie niezależnych. Im więcej tych punktów, tym lepsza grafika (większa rozdzielczość). Na ekranie można więc rysować różne krzywe, rzuty perspektywiczne, a nawet tworzyć dzieła sztuki. Dużą lub małą rozdzielczość da się porównać z pracą dwóch malarzy: pierwszy tworzy za pomocą cienkiego pędzelka (duża rozdzielczość), a drugi stosuje szeroki pędzel ławkowy (mała rozdzielczość).

GRAFIKA BARWNA

Prawie wszystkie mikrokomputery dają do dyspozycji od 8 do 16 kolorów. Poszczególne kolory mogą się podobać lub nie, w zależności od gustu. W większości komputerów możliwa jest również zmiana intensywności barw. W ten sposób można uzyskać nawet do 128 różnych „kolorów". Czy wszystkie one są Wam rzeczywiście potrzebne? Musicie sobie sami odpowiedzieć na to pytanie. Jeśli barwy odgrywają istotną rolę w Waszych planach wykorzystania mikrokomputera, to liczba kolorów jest istotna. Zbadajcie sposób korzystania z poszczególnych kolorów i łatwość posługiwania się instrukcjami związanymi z operowaniem barwą. Decyzji o zakupie nie podejmujcie jedynie na podstawie liczby i odcieni kolorów oferowanych przez producenta. Ważne jest raczej to, co chcecie osiągnąć na tym sprzęcie. Przed dokonaniem wyboru nie żałujcie czasu na zdefiniowanie stawianych komputerowi wymagań.

OPROGRAMOWANIE

Aby komputer mógł wypełniać określone zadania, potrzebne są odpowiednie programy składające się na jego oprogramowanie. Przy źle wybranym oprogramowaniu, zamiast pożytecznej maszyny będziecie mieli tylko zbędny mebel zbierający kurz.

Na ogół dla każdego komputera domowego opracowany jest duży zestaw programów – od gier telewizyjnych, poprzez warcaby i szachy aż do komputerowego planowania jadłospisu, od zadań rachunkowych dla dzieci aż do planowania zużycia energii w domu. Powstaje pytanie, jak wejść w posiadanie tych programów oraz ile każdy z nich kosztuje.

Dla tych, którzy jeszcze nie potrafią lub nie chcą samodzielnie pisać programów, istnieje znaczna liczba programów gotowych. Trzeba tylko mieć świadomość, że programy opracowane dla danego typu komputera nie będą współpracowały z innymi! Jeśli nie natraficie na właściwy program, zapisany już na dyskietce lub taśmie kasetowej, spróbujcie poszukać go w opracowaniach coraz częściej publikowanych w czasopismach i książkach. Znalezione tam programy można samemu wprowadzić do pamięci w nie zmienionej wersji lub też odpowiednio przystosować do własnych potrzeb.

PAMIĘĆ ZEWNĘTRZNA

Mówiliśmy już o pamięci RAM i pamiętamy, że po wyłączeniu komputera jej zawartość jest niszczona. Z tego powodu dane, na których nam zależy, a które znajdują się w RAM, należy przed wyłączeniem komputera zapisać w takim miejscu, gdzie nie ulegną one zniszczeniu. Miejsce to nazywa się *pamięcią zewnętrzną*. Do wyboru mamy dwie możliwości mieszczące się w możliwościach finansowych przeciętnego użytkownika: magnetofon kasetowy lub stację dysków elastycznych, zwanych dyskietkami. Obsługa magnetofonu jest taka sama jak w przypadku kaset z muzyką: programy lub dane z komputera nagrywane są na taśmę lub przegrywane z powrotem do komputera. Dyskietki wykorzystuje się w podobny sposób, z tą różnicą, że informacje zapisywane są nie na taśmie, lecz na pokrytym warstwą magnetyczną nieco grubszym od taśmy krążku, umieszczonym w ochronnej kopercie z plastiku.

Podstawowa różnica leży w cenie, ponieważ stacja dysków elastycznych kosztuje o wiele więcej niż magnetofon kasetowy. Z drugiej strony czas dostępu do danych zapisanych na dyskietkach wynosi tylko ułamek sekundy, podczas gdy odtworzenie lub zapis odpowiednich danych na taśmie może trwać kilka minut.

O CZYM JESZCZE TRZEBA PAMIĘTAĆ?

Jedną z bardzo ważnych rzeczy, niezbędnych do pracy z komputerem, jest dobry podręcznik. Powinny się w nim znaleźć wszystkie bez wyjątku instrukcje i funkcje. Podręcznik ma Was prowadzić, pomagać w pracy, udzielać wyczerpujących i pomocnych wskazówek przy ewentualnie napotkanych błędach. Dobry i dokładny opis umożliwia efektywną pracę i pomoże znaleźć nowe dziedziny zastosowań komputera.

Komputer nie potrafi myśleć. Jakość jego pracy jest taka, jakie są wprowadzone do niego dane i programy. Jeśli otrzyma fałszywe lub złe dane, poda zły lub fałszywy wynik. Istnieje międzynarodowe określenie takiego stanu rzeczy: GIGO (ang. garbage in – garbage out) co po polsku znaczy: wprowadzisz śmieci – śmieci otrzymasz.

Pamięci... zapominają! Nie często wprawdzie, ale jak uczy doświadczenie, w najbardziej niewłaściwych momentach. Raz może się to objawić zawieszeniem działania poprawnego programu, innym razem zawodzi system operacyjny. Cóż począć? Uruchamiamy wtedy program powtórnie, odtwarzamy z dyskietki system operacyjny. Ale co się stanie, gdy z powodu błędu bloku pamięci otrzymamy fałszywe wyniki obliczeń, brane przez nas, wobec braku jakiejkolwiek sygnalizacji, za poprawne? Lepiej zajmijmy się więc metodami, które pozwalają wyeliminować, a przynajmniej ograniczyć ryzyko błędnego odczytu zawartości pamięci.

Istnieje wiele sposobów wykrywania błędów w systemach cyfrowych. Ogólnie można powiedzieć, że polegają one na rozszerzaniu słowa danych o pewną liczbę bitów, stanowiących kontrolną część przesyłanego słowa. Bity te, zwane *bitami kontrolnymi* są generowane automatycznie na podstawie postaci słowa danych według z góry określonego algorytmu (kodu). Kody, które pozwalają wykryć zaistniały w systemie błąd, nazywamy *kodami detekcyjnymi*. Powstanie kodów detekcyjnych nie zakończyło jednak prac nad problematyką wykrywania błędów w układach cyfrowych. Następnym etapem było opracowanie takich kodów, które umożliwiają nie tylko wykrycie błędów, lecz także ich automatyczne eliminowanie. Kody takie, stanowiące dziś podstawę działania specjalizowanych układów VLSI (np. Intel 8206) określamy mianem *kodów detekcyjno-korekcyjnych*.

Aby przybliżyć ideę detekcji błędów, rozważmy najprostszy sposób jej realizacji, a mianowicie generację tzw. *bitu parzystości*.

Niech nasze kontrolowane słowo danych ma długość czterech bitów. Przyjmijmy, że do takiego słowa będziemy dodawać bit parzystości, tworzony według następującej reguły: jeżeli w słowie danych występuje parzysta liczba jedynek, to bit parzystości przyjmuje wartość logiczną 1, w przeciwnym razie 0. Tak powstałe pięciobitowe słowo zostaje wpisane do pamięci. Podczas odczytu danej z pamięci, bit parzystości jest tworzony po raz drugi. Wystarczy teraz porównać oba słowa, aby uzyskać informację, czy wystąpił błąd (tab. 1).

Zaraz, zaraz! – zakrzykną wnikliwi czytelnicy. A co się stanie, gdy wystąpi podwójny błąd przy odczycie? (tab. 2).

Oczywiście, porównywanie bitów parzystości nie daje pełnej gwarancji poprawności

Tabela 1

	Dane				Bit parzystości
	b0	b1	b2	b3	b4
Zapis	0	0	0	0	1 bity takie same, nie ma błędu
Odczyt	0	0	0	0	1
Zapis	0	0	1	1	1 bity różne, błąd!
Odczyt	0	0	0	1	0

Pamięci na cenzurowanym

odczytu. Jednak prawdopodobieństwo wystąpienia podwójnego błędu (ogólnie błędu na parzystej liczbie pozycji słowa danych) jest nieporównywalnie mniejsza niż prawdopodobieństwo pojedynczego przekłamania. Nasz system, aczkolwiek niedoskonały, jest już znacznie pewniejszy.

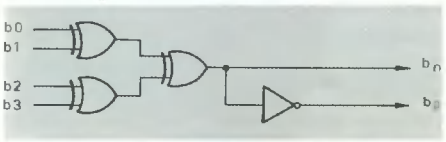
Tabela 2

Dane					Bit parzystości
Zapis	1	1	1	1	1 bity takie same mimo błędu
Odczyt	0	0	0	0	1

Generowanie bitu parzystości dla k-bitowego słowa danych polega na obliczaniu wartości wielomianu binarnego o następującej postaci:

$b_p = b_0 + b_1 + b_2 + \cdots + b_{k-2} + b_{k-1}$

gdzie + oznacza operację Ex-OR (suma modulo 2). Fizycznie realizuje się to przez odpowiednie połączenie bramek typu 7486 (rys. 1 – dla czterech bitów). Na wyjściu b_n otrzymujemy stan logiczny 1 wtedy, gdy nieparzysta liczba wejść jest w stanie logicznym 1. Bit ten, będący zawsze negacją bitu parzystości, jest określony jako *bit nieparzystości.*



Rys. 1. Generowanie bitów parzystości i nieparzystości

W serii układów TTL 74LS jest realizowany scalony układ generowania bitu parzystości, oznaczony numerem 74LS280, będący poprawioną wersją starszego układu 74180. Układ ma 9 wejść oznaczanych literami A...I oraz wyjście bitu parzystości i nieparzystości. Generator działa według zasady przedstawionej w tab. 3.

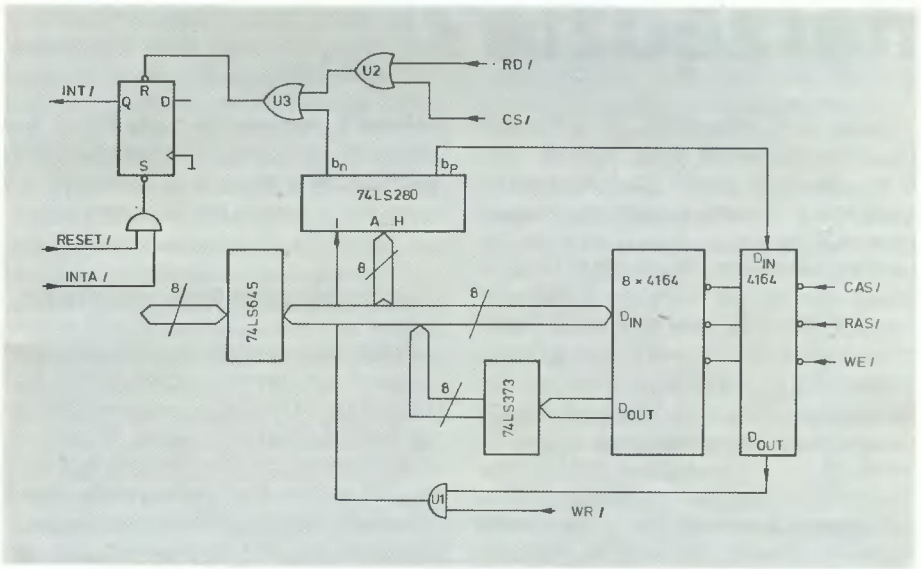
Kostka 74LS280 stanowi podstawę realizacji większości układów detekcji błędów w blokach pamięciowych. Zasadę użycia tego układu w bloku pamięci o organizacji 64Kx8 przedstawiono na rys. 2.

Na układ składają się: blok pamięci o organizacji 64Kx9 zbudowany z kostek 4164, układ generacji i kontroli parzystości – układ 74LS280 oraz przerzutnik przerwań – przyjęto bowiem, że w wypadku wystąpienia błędu zostanie wygenerowane przerwania.

Tabela 3

Liczba wejść w stanie 1	Bit parzystości	Bit nieparzystość
0, 2, 4, 6, 8	1	0
1, 3, 5, 7, 9	0	1

Działanie układu jest następujące. Zapisywane do pamięci słowo pojawia się na wejściach bloku pamięci oraz układu 74LS280. Układ ten generuje bit parzystości na podstawie liczby jedynek w słowie danych, bowiem na dziewiątym wejściu sygnał WR/ wymusza poprzez bramkę U1 stan logiczny Ø.



Rys. 2. Blok pamięci z kontrolą parzystości

Tak powstały bit parzystości jest zapisywany do dziewiątej kostki 4164 równolegle ze słowem danych. W czasie odczytu sygnał WR/ ma poziom wysoki, czyli do wejść układu 74LS280 są doprowadzane zarówno dane, jak i wygenerowany przy zapisie bit parzystości. Na jego wyjściach pojawia się więc nowy stan utworzony na podstawie wszystkich dziewięciu bitów. Zwróćmy uwagę, że przy braku błędu wśród tych dziewięciu bitów będzie zawsze nieparzysta liczba jedynek. Przy prawidłowym odczycie wyjścia b_p będzie w stanie Ø, a b_n w stanie 1. Gdy wystąpi błąd, na wejściach bramek U3 pojawią się jednocześnie stany niskie i w konsekwencji przerzutnik zostanie wyzerowany, powodując przerwanie. Sygnał przyjęcia przerwania (INTA) ustawi linię INT/ znów w stan wysoki.

Oczywiście, aby układ funkcjonował prawidłowo, przerzutnik musi być odpowiednio ustawiony sygnałem RESET/, podczas włączania komputera. Od projektanta systemu zależy działanie podjęte po przyjęciu przerwania; najczęściej procedura ograniczy się do wysłania komunikatu o błędzie.

Po włączeniu zasilania zawartość pamięci jest przypadkowa – może to być powodem zgłoszenia przerwania przy próbie odczytu. Należy zatem rozpocząć od zablokowania przerwań i zapisania całego bloku dowolną daną, najczęściej zerami. Spowoduje to właściwe ustawienie wszystkich bitów parzystości w pamięci, dopiero wtedy można odblokować system przerwań.

Proponowany układ, aczkolwiek nie będący ofertą dla początkujących konstruktorów, z pewnością pozwoli zmniejszyć liczbę siwych włosów na (młodych) głowach konstruktorów i użytkowników mikrokomputerów.

Polsko-angielski słowniczek terminów

parzystość – *even parity*

nieparzystość – *odd parity*

układ generacji i kontroli bitu nieparzystości i parzystości – *odd/even parity generator/checker*

bit parzystości – *parity bit*

detekcja i korekcja błędów – *error detection/correction* (EDC)

kody korekcyjne – *error correction codes* (ECC)

JERZY ORKISZEWSKI

Luksus czy pożyteczne narzędzie?

Każdy szef odpowiada za całokształt przedsięwzięć swojej firmy, zakładu bądź działu. Ich obraz tworzy przez wystąpienia publiczne, a zwłaszcza wszelkiego rodzaju pismami przesyłanymi do partnerów w działalności gospodarczej. Każdy list to karta wizytowa, kształtująca wyobrażenia partnerów o jego autorze. Na podstawie treści i formy pisma kształtują oni własny, często bardzo subiektywny i... prawodopodobnie nieprawdziwy obraz. Dlatego szef pragnie szczególnie starannie redagować swe pisma, co wcale nie jest łatwe. Jego sekretarka stara się pisać te pisma bez błędów, by uniknąć późniejszych poprawek, zaś najważniejsze fragmenty tekstu odpowiednio wyróżnić. Szef stara się również mało przejrzyste zależności przedstawiać w postaci łatwo zrozumiałych zestawień i wykresów. Czyni wszystko, aby jego pisma wyróżniały się korzystnie w powodzi innych dokumentów.

DIAGRAMY Z MIKROKOMPUTERA

Grafika jest przydatna nie tylko w korespondencji, ale również w czasie obrad i dyskusji. O ileż łatwiej jest pojąć współpartnerom złożone zależności, jak np. kształtowanie się obrotów, zysków i kosztów za ostatni rok, gdy zamiast mało przejrzystych kolumn liczbowych zobaczą oni kolorowe diagramy. Wykresów tych szef nie musi zlecać do wykonania specjalnie zatrudnionemu grafikowi, gdyż pracę wykona mikrokomputer, na podstawie wprowadzonych danych początkowych i żądanej formy wyników.

Dalszą pomoc stanowią zbiory i bazy danych. Sekretarka na ogół prowadzi obszerną kartotekę. Ileż pracy wymaga przygotowanie szefowi wyciągów uporządkowanych według różnych kryteriów! Jeśli zawartość kartoteki zostanie raz wprowadzona do mikrokomputera, będzie tam zapamiętana na stałe w postaci zbiorów danych. Korzystając z tych zbiorów, sekretarka może w każdej chwili (bez pomyłek!) wydrukować szefowi potrzebne informacje, posortowane według dowolnie wybranego kryterium, może też na żądanie wyszukać pojedynczą informację. Jeśli więc np. sekretariat prowadzi kartotekę klientów i dostawców, to szef na każde życzenie otrzyma wyciągi uporządkowane według nazwisk, adresów, terminów spotkań, dat urodzin, obrotów, przeterminowanych należności itp.

DZIEŃ URODZIN ZE ZBIORU DANYCH OSOBOWYCH

Na podstawie danych osobowych sekretariat może np. co miesiąc przygotowywać szefowi listę współpracowników obchodzących w ciągu najbliższych czterech tygodni imieniny lub jubileusze. Nie zapomni on wtedy o złożeniu im życzeń. Zbiory danych osobowych zawierają też adresy oraz tytuły, jakich należy użyć przy pisaniu indywidualnie kierowanych ofert lub listów z życzeniami. Mikrokomputer pisze te teksty automatycznie, według odpowiednich wzorców, oszczędzając sekretarce czas, który będzie mogła przeznaczyć na inne, ważniejsze prace. Sekretarka może też, jeśli szef nie chce robić tego osobiście, przeprowadzać kalkulacje i wyliczać koszty. Służą do tego wzory matematyczne umieszczone w programach.

TERMINARZ W KOMPUTERZE

Szef może za pomocą mikrokomputera prowadzić terminarz. Codziennie otrzymuje wtedy wydruk harmonogramu dnia, zawierający nie tylko nazwiska rozmówców i uzgodnione terminy spotkań, lecz także tematy rozmów i istotne dla tych rozmów okoliczności. Szczególną korzyść osiąga się przy prowadzeniu terminarza dla wielu współpracowników. Komputer zastępuje wtedy sekretarkę w bardzo pracochłonnym uzgadnianiu terminu spotkania odpowiadającego wszystkim partnerom.

DYKTOWANIE NA EKRAN

Niezłą pomocą jest też mikrokomputer przy zazwyczaj bardzo kłopotliwym zajęciu: dyktowaniu tekstu dla maszynistki. Wielu szefów niechętnie stosuje w tym celu dyktafon, ponieważ trzeba do niego mówić przemyślane zdania, których nie można już potem zmienić. Przy dyktowaniu maszynistce wpisującej tekst do komputera, można obserwować go na ekranie, w dowolnym miejscu przerwać wprowadzanie i poprosić o zmianę już wprowadzonych informacji. Możliwe są również poprawki stylistyczne w już napisanych listach, gdyż tekst zapamiętany na dyskietce w każdej chwili może być wyświetlony na ekranie i zmodyfikowany. W ten sposób praca sekretarki staje się lżejsza i jest wykonywana szybciej.

TEKSTY Z CEGIEŁEK

Niewielu szefów wie, jak często ich pracownice muszą stale od nowa wystukiwać tak samo brzmiący tekst. Nie chodzi tu tylko o w pełni identyczne listy, ale wciąż powtarzające się w korespondencji sformułowania takie jak: „Wielce szanowny Panie!" i „Łączę wyrazy szacunku", czy też zawierający często wiele znaków podpis. Bardzo pracochłonne są również powtarzane często opracowania redakcyjne skomplikowanych tekstów oraz ofert, w których zmianie ulegają tylko niewielkie ich fragmenty. W wielu listach tekst pozostaje całkowicie niezmieniony, zmienia się tylko adres. Ten, może sekretarka dopisać automatycznie na podstawie wcześniej wprowadzonych danych, wystukując na klawiaturze tylko identyfikator w postaci dwu- lub trzycyfrowej liczby.

Przy innych wiadomościach, na przykład pierwszym, drugim i trzecim ponagleniu, czy też przy wielu ofertach, tak samo brzmiące fragmenty tekstu muszą być połączone w różny sposób. Fragmenty te sekretarka opatruje identyfikatorami i umieszcza w „katalogu tekstów", którego zawartość wprowadza do komputera. Potem wystarczy, że szef powie odpowiednią liczbę, a sekretarka będzie ją mogła wystukać na klawiaturze zamiast tekstu. Komputer odszuka w pamięci odpowiedni fragment tekstu i wyświetli go na ekranie jako część właśnie przygotowywanego pisma.

MIKROKOMPUTER Z JEDNYM CZY Z WIELOMA TERMINALAMI?

Do prowadzenia wszystkich prac w zespole szef-sekretarka wystarcza w zasadzie jeden mikrokomputer. Niekiedy jednak oboje chcą miec jednocześnie dostęp do danych i możliwości obliczeniowych. Mogą wtedy zainstalować dwa systemy jednodostępne, z jednym terminalem każdy, lub jeden system z dwoma terminalami. W obu wypadkach zarówno szef, jak i sekretarka dysponują własnym, oddzielnym monitorem ekranowym i klawiaturą. W urządzeniach jednodostępnych każdy dysponuje komputerem z własną pamięcią. Wymiana danych następuje przez wymianę dyskietek lub połączenie obu komputerów w sieć. W systemie wielodostępnym szef i sekretarka wykorzystują ten sam komputer centralny i jego pamięć. Mają więc dostęp do tych samych danych i to zawsze aktualnych. Przy urządzeniach jednodostępnych każdy komputer może (choć nie musi) mieć własną drukarkę. W systemie wielodostępnym używa się tylko jednej drukarki centralnej, co znacznie zmniejsza koszt systemu, bowiem dobre drukarki są bardzo drogie.

JAKIE WYPOSAŻENIE?

W mikrokomputerze niezbędny jest ekran. Powinien on dać możliwość wyświetlania 80 znaków w wierszu. Nie może więc to być zwykły ekran telewizyjny o wierszach długości 40 lub 32 znaków i możliwości wyświetlania jednorazowo tylko 20 wierszy. Monitor alfanumeryczny (wyswietlający tylko znaki i pozbawiony możliwości graficznych) w pracy sekretariatu jest mało przydatny. Trudno rozstrzygnąć, czy obraz powinien być monochromatyczny (jednobarwny) czy kolorowy. Zależy to od tego, do czego urządzenie ma być wykorzystywane. Do przetwarzania tekstów szczególnie nadaje się ekran monochromatyczny. Jego wyrazistość jest na ogół znacznie większa od ekranu barwnego, można więc przy nim długo pracować bez znużenie. Istnieją już wprawdzie ekrany barwne dorównujące rozdzielczością monochromatycznym – lecz są one bardzo kosztowne. Kolor iest wyrazem nowoczesności, ale szczególnie cenny tylko w zastosowaniach graficznych, które również i w wersji jednobarwnej mogą być przekonywające. Barwny ekran ma sens w zasadzie tylko wtedy, gdy również na drukarce można otrzymać barwne wydruki.

Co do klawiatury, właściwie wybór jest jeden: musi to być klawiatura zawierająca specyficzne dla używanego języka znaki i zgodna ze znormalizowaną klawiaturą maszyny do pisania.

Podstawowe znaczenie w sekretariacie ma pamięć zewnętrzna. Najbardziej odpowiednie są tu dyski elastyczne zwane dyskietkami. Umożliwiają one znacznie szybszą pracę niż kasety magnetyczne. Mikrokomputer musi mieć koniecznie dwa napędy dyskietek, ponieważ przy jednym napędzie obsługa jest kłopotliwa i czasochłonna. Współczesne dyskietki mają pojemność od 160 KB do 1,2 MB (1 KB = 1 kilobajt = 1024 bajty, czyli znaki alfanumeryczne, 1 MB = 1 megabajt = 1024 KB). Najlepsze są mikrokomputery wyposażone, oprócz dysku elastycznego, dodatkowo w sztywny dysk o pojemności 10-40 MB. Dla komputera zainstalowanego w sekretariacie jest to kombinacja idealna, choć nie konieczna.

PIĘĆ TYPÓW DRUKAREK

Wybor drukarki nie jest sprawą prostą, ponieważ dobra drukarka kosztuje obecnie więcej niż mikrokomputer. Oszczędność byłaby tu jednak błędem, zwłaszcza w sekretariacie, w którym drukuje się reprezentacyjne pisma i grafiki. W zasadzie jest do wyboru pięć różnych typów drukarek: rozetkowa, mozaikowa, uniwersalna (graficzno-tekstowa), strumieniowa i elektrooptyczna (najczęściej laserowa). Drukarki rozetkowe i mozaikowe wykonują wydruki niezłej jakości: drukarka rozetkowa przy przetwarzaniu

Sekretarka za pomocą mikrokomputera jednostanowiskowego prowadzi całą korespondencję, aktualizuje zbiory danych oraz przekształca w postać graficzną złożone zestawienia liczbowe

tekstów, natomiast drukarka mozaikowa przy przetwarzaniu danych i grafice. W wielu zastosowaniach jakości ich wydruków nie można już niestety uznać za wystarczającą. Droższe, lecz bardziej wszechstronne są inne typy drukarek. Drukarka uniwersalna jest urządzeniem stosującym matrycę igłową. Dzięki temu może ona np. drukować wytłuszczone (pogrubione) wyrazy. Drukarki strumieniowe i elektrooptyczne nie tyle drukują, ile bezszmerowo „malują" grafiki i teksty. Drukarki wyższej klasy pracują bardzo szybko. Drukarki strumieniowe i elektrooptyczne mogą drukować tylko w jednym egzemplarzu (bez kopii). Drukarki elektrooptyczne nadrabiają to jednak szybkością, dostarczając wymaganą liczbę egzemplarzy w krótszym czasie, niż drukarki wykonujące kopie.

Kopie pisane przez kalkę nie są dziś tak ważne ze względu na możliwość wykonania foto- i kserokopii. Mogą natomiast powstawać problemy przy kopiowaniu formularzy, szczególnie, gdy ze względu na obieg dokumentów, poszczególne kopie drukowane są na różnych kolorach papieru, lub gdy kopie zawierają tylko część informacji zawartych w oryginale.

Przy użyciu dowolnego z omówionych trzech typów drukarek wysokiej klasy sekretarka może drukować pisma, używając różnych rodzajów czcionki i stosując dowolne rozmieszczenie tekstu, bez konieczności przerywania pisania dla zmiany głowicy drukującej. Drukarki wysokiej klasy pozwalają więc uzyskać pisma prezentujące się szczególnie estetycznie. Jest to cenne dla sekretariatu, w którym przywiązuje się dużą wagę do spraw reprezentacji.

AUTOMATYCZNY PODAJNIK PAPIERU

Wkręcanie papieru jest czynnością czasochłonną. Lepsze modele drukarek wyposażone są w pół- lub w pełni zautomatyzowane urządzenia podające papier. Pełne automaty mogą same w odpowiednim momencie zakładać do drukarki pierwszą lub następną stronę (ew. rożniącą się nagłówkiem) oraz kopertę. Inne urządzenia ułatwiają drukowanie etykiet.

PROGRAMY I KOMPATYBILNOŚĆ SPRZĘTU

Kazdy komputer pracuje pod nadzorem systemu operacyjnego. Komputery 8-bitowe wykorzystują najczęściej system CP/M, zaś 16-bitowe MS-DOS lub podobny. Te szeroko rozpowszechnione systemy umożliwiają korzystanie z wielkiej liczby współpracujących z nimi programów użytkowych. Jest to sprawa zasadnicza, bo program to przecież dusza komputera. Mikrokomputer bez oprogramowania nie ma żadnej wartości użytkowej.

Szczególnie istotna w dziedzinie mikrokomputerów jest kompatybilność, czyli zgodność jednych instalacji z innymi. Umożliwia ona bez problemów stopniową rozbudowę systemu. Używanie identycznych modeli zapewnia ich pełną wymienność. Nośniki danych, klawiatury, obsługa i programy są wtedy takie same. Łączenie różnych modeli jest często także możliwe, co powinno być jednak potwierdzone przez dostawcę. Warto sobie uświadomić, że chęć zachowania kompatybilności wiąże nabywcę z określonym producentem, oraz że kompatybilne wyroby mogą w danej chwili już być przestarzałe pod względem technicznym. Dlatego przed zakupem komputera należy zorientować się w możliwościach zastąpienia go przez inny sprzęt. Aby uniknąć rozczarowań, warto też żądać zademonstrowania działania sprzętu.

MIKROKOMPUTER Z TELETEKSTEM I TELEKSEM

Przy sprzedaży mikrokomputerów oferowane są na ogół dodatkowe opcje: możliwość przyłączenia do sieci komputerowej, teletekst oraz teleks.

Przyłączenie do sieci mikrokomputerowej umożliwia wymianę informacji z innymi komputerami oraz dostęp do ich baz danych. Teletekst daje wiele korzyści, konieczny jest jednak (ewentualnie jako urządzenie dodatkowe) telewizor kolorowy. Teleks jest wtedy istotny, jeśli biuro prowadzi ożywioną korespondencję z partnerami posiadającymi dalekopisy.

TAJEMNICA SUKCESU: PODWÓJNY SEKRETARIAT

Niezakłócony rytm pracy powinien być także wspomagany przez przedsięwzięcia organizacyjne. Szczególnie dobrze sprawdziły się podwójne sekretariaty. Ten sposób organizacji pracy można stosować jedynie wówczas, gdy każda sekretarka pracuje nie tylko dla jednego szefa. Dla lepszego rytmu pracy konieczne jest, by w danej chwili tylko jedna z dwóch sekretarek zajmowała się telefonami i szefami.

Jeśli zakład ma odnieść odczuwalną korzyść z racjonalizacji, nie wystarczy zainstalowanie tylko jednego mikrokomputera. To nowoczesne urządzenie okaże się jednak przedwczesnym zakupem, jeśli teksty do drukowania będą przygotowane nadal w postaci rękopisów bądź stenogramów.

micro

Nowe komputery – BBC Master Series

Komputery serii BBC, produkowane przez firmę Acorn, są w Polsce mało znane. Na rodzimym, brytyjskim rynku cieszyły się jednak zawsze dużą popularnością. Stosowane są głównie w edukacji – dość wspomnieć, że siedem z dziesięciu komputerów w angielskich szkołach to właśnie BBC model B.

Nie wszystkie jednak produkty firmy Acorn były udane. Następca BBC B, Acorn Electron, nie był dobrze przyjęty przez rynek. Istotną przeszkodą była również jego wysoka cena – około 500 funtów. W tym czasie pojawiło się wiele innych komputerów, w tym o połowę tańsze i bogatsze w możliwości Amstrady. Anglicy czekali z niecierpliwością – co na to Acorn?

I oto ostatnio karty zostały odkryte. Acorn zaanonsował nie jeden model, lecz całą serię nowych maszyn pod nazwą Master Series.

W skład serii wchodzą: MS-128 (jak zwykle końcowy numer odpowiada pojemności pamięci RAM w KB), jego szybsza wersja MS-128 TURBO, MS-512 – 16-bitowy komputer biurowy, MS-Scientific – 32-bitowy komputer przeznaczony do zastosowań naukowych i wreszcie MS-ET – uproszczony model mający spełniać rolę terminala w sieci Econet.

Najprostszy model ma 128 KB pamięci RAM. Potwierdza to tezę, że Amstrad 6128 ustalił poziom, poniżej którego nie wypada już schodzić. Zresztą postępy w technologii jakby same narzucały takie rozwiązania. Cała pamięć operacyjna MS-128 to cztery kostki dynamiczne o organizacji 64 K × 4. Poza nimi – procesor 65C12, będący 2 MHz wersją „starego, poczciwego" 6502, ROM o pojemności 128 KB z monitorem, dwoma systemami operacyjnymi i szeregiem procedur aplikacyjnych oraz sześć specjalizowanych sterowników zbudowanych w oparciu o matryce logiczne. Są to sterowniki: pamięci RAM, szyny peryferyjnej, monitora ekranowego, klawiatury, matrycy kolorów i układów we-wy.

MS-128 ma liczne porty umożliwiające znaczną rozbudowę systemu. Z tyłu komputera umiejscowiono: złącze sieci Econet, wyjście na wzmacniacz akustyczny, port magnetofonu, we-wy analogowe, wyjścia RGB do sterowania monitorem kolorowym, wyjście UHF dające możliwość wykorzystania telewizora i w końcu wyjście zespolonego sygnału wizyjnego dla monitora monochromatycznego.

Dodatkowo, od spodu komputera (?!), dołączyć można napędy dyskietek i drukarkę. Są również wyprowadzone dwie szyny systemowe: jedna przeznaczona do rozszerzenia systemu o dysk sztywny Winchester, druga – daje możliwość dołączenia drugiego procesora. Może to być na przykład Z80, co umożliwia instalację systemu operacyjnego CP/M.

Komputery Master Series rozpoczynają podbój Wielkiej Brytanii. A kiedy dotrą do Polski?

Superlife

Symulacja rozwoju żywych komórek

Gra symulująca rozwój żywych komórek powstała ok. 20 lat temu. Proste zasady i niewątpliwe walory estetyczne przyczyniły się do jej dużej popularności.

Program dla ZX Spectrum 48K powstał w połowie 1984 r. Głównym zadaniem, które sobie postawiłem, było wykorzystanie w pełni możliwości graficznych ZX Spectrum i szybkości procesora Z80A. Program pozwala obserwować rozwój kolejnych pokoleń na planszy 256 × 176. Czas życia kolejnych pokoleń waha się od 0,5 do 1,3 s, w zależności od obszaru zajętego przez komórki.

OGÓLNE ZASADY SYMULACJI

W grze zastosowano bardzo proste reguły symulacji rozwoju komórek; są one bliższe matematyce niż biologii. Symulacja odbywa się na płaszczyźnie podzielonej na przystające kwadraty. Po ustaleniu układu początkowego (pokolenie zerowe), tzn. zapełnieniu skończonej liczby kwadratów pojedynczymi komórkami, każde pokolenie *N+1* powstaje z pokolenia *N* według następujących zasad:

1. Komórka „umiera z samotności", jeśli w pokoleniu *N* z sąsiednich ośmiu pól, mniej niż dwa pola były zajęte przez komórki;
2. Komórka „umiera z przeludnienia", jeśli w pokoleniu *N* miała więcej niż trzech sąsiadów;
3. W wolnym polu „rodzi się" komorka, jeśli w pokoleniu *N* pole to ma dokładnie trzy sąsiednie pola zajęte przez komórki.

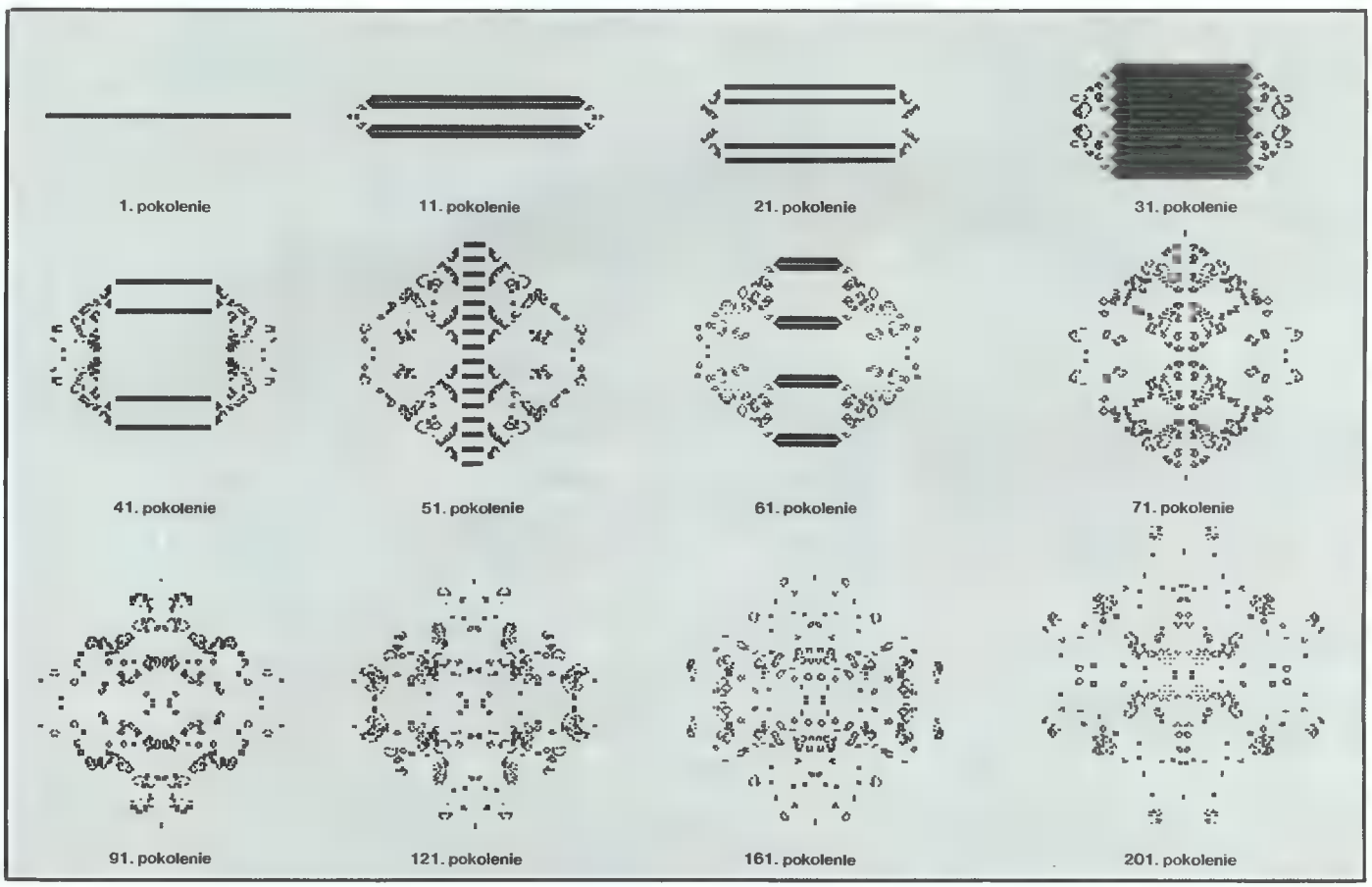
WPROWADZENIE PROGRAMU

Do wprowadzenia programu nie jest potrzebna umiejętność programowania. Przydatna jest natomiast elementarna znajomość obsługi ZX Spectrum. Należy postępować ściśle według podanego algorytmu:

1. Wprowadź program SUPERLIFE napisany w języku BASIC. Zwróć szczególną uwagę na linię *4* (po REM trzeba wprowadzić 343 znaki).
Uwaga! W linii 220 napis INK to trzy duże litery, a nie instrukcja INK.
2. Po zakończeniu wpisywania programu wprowadź polecenie *LIST*, następnie naciśnij klawisz *BREAK*, a potem jednoczesnie klawisze *CAPS SHIFT* i *1*. Linia numer *4* powinna znaleźć się w dolnej części ekranu. Zmień numer linii z *4* na *3* i wciśnij ENTER.
3. Obecnie program zawiera również linię *3*, identyczną z linią *4*.

Powtórz tę operację w celu uzyskania linii *1* i *2*, identycznych z linią *4*.

4. Zapisz program jako wersję pośrednią programu ostatecznego.
5. Aby wpisać kod maszynowy, należy rozpocząć realizację programu przez podanie polecenia *GOTO 2000*. Na ekranie pojawi się liczba *23760*. Należy wówczas wpisać cały wiersz cyfr i liter (16 pozycji). Po zakończeniu wpisywania wiersza należy nacisnąć klawisz *ENTER*. Jeśli wyświetlona obok suma kontrolna będzie inna od sumy podanej w artykule, należy nacisnąć klawisz *n* oraz *ENTER* i ponownie (bez błędu!) wprowadzić 16 znaków. Jeśli suma jest prawidłowa należy wcisnąć klawisze *t* i *ENTER*.
6. Zostanie podany kolejny adres *23768*. Postępujemy jak w poprzednim punkcie. Opisane czynności powtarzamy, aż zostanie wpisany cały kod zamieszczony na wydruku.
7. Po zakończeniu wpisywania należy uruchomic nagrywanie na magnetofon i zarejestrować całość programu.
8. Nagranie można powtórzyć przez naciśnięcie klawisza *BREAK* i wpisanie komendy *GOTO 2260*.



OBSŁUGA PROGRAMU

Naciśnięcie klawisza *d* powoduje przejście programu do trybu demonstracyjnego. Kolejno wyświetlane są (linie od 1000 do 1240) cztery różne typy układów początkowych komórek. Przejście do następnego układu początkowego następuje po naciśnięciu klawisza SPACE.

Przyciśnięcie klawisza *w* po rozpoczęciu realizacji programu daje możliwość zapisania dowolnego układu początkowego i obserowowania jego rozwoju.

Ze względu na ograniczoną pojemność dwóch wierszy u dołu ekranu są tam podawane tylko niezbędne skróty: kolor komórek *0* lub *7* oraz współrzędne *X, Y* kursora.

Dostępne klawisze

5 – ruch kursora w lewo

■ – ruch kursora w dół

7 – ruch kursora w górę

8 – ruch kursora w prawo

p – ruch kursora w górę i prawo

q – ruch kursora w górę i lewo

SPACE – ruch kursora w dół i prawo

z – ruch kursora w dół i lewo

0 – zmiana koloru komórek

ENTER – rozpoczęcie generowania kolejnych pokoleń.

Podczas generowania kolejnych pokoleń jest dostępny jedynie klawisz *SPACE* (nie *BREAK*!). Po około sekundowym okresie wciśnięcia tego klawisza program przestaje generować następne pokolenia. Dostępne są wówczas następujące możliwości:

x – kasowanie całego ekranu

c – kontynuacja rozwoju

z – możliwość modyfikacji układu na ekranie.

Szczególnie ciekawe są układy powstające z bardzo prostych układów początkowych, jak np. pasek o długości od 80 do 150 komórek.

Program nie działa na komputerze ZX Spectrum 16K.

JERZY SZAFRANEK

Wydruk 1. Program SUPERLIFE

```
   4 REM xxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
  10 REM
  13 REM ■■■u
  20 POKE 23800,0: CLS
  30 PRINT " J.Szafranek""" 'S
u p e r l i f e' "
  40 PRINT "_____
_____"
  50 PRINT AT 10,5;"d-DEMONSTRAC
JA"
  60 PRINT AT 12,5;"w-WPROWADZEN
IE DOWOLNEGO          UKLADU"
  70 LET r$=INKEY$: IF INKEY$=""
 THEN  GO TO 70
  ■■ IF r$="w" THEN  GO TO 150
  90 IF r$="d" THEN  GO TO 1000
 100 GO TO 70
 150 FOR i=1 TO 100: NEXT i
 160 CLS
 200 REM
 203 REM wprowadzanie ukladu
 210 LET x=128: LET y=88: LET z=
0
 220 PRINT #1;AT 1,1;"INK=";7*z;
"   X=";x;"  ";AT 1,16;"Y=";y;"
 ";AT 1,25;
 230 LET r$=INKEY$: PLOT  OVER
ABS (z-1);x,y: PAUSE 10: PLOT  ■
VER z;x,y: PAUSE 10: IF r$="" TH
EN  GO TO 230
 240 IF r$="5" THEN  LET x=x-1:
GO TO 220
 250 IF r$="8" THEN  LET x=x+1:
GO TO 220
 260 IF r$="6" THEN  LET y=y-1:
GO TO 220
 270 IF r$="7" THEN  LET y=y+1:
GO TO 220
 280 IF r$="0" THEN  LET z=ABS (
z-1): GO TO 220
 290 IF r$="q" THEN  LET y=y+1:
LET x=x-1: GO TO 220
 300 IF r$="p" THEN  LET y=y+1:
LET x=x+1: GO TO 220
 310 IF r$=" " THEN  LET y=y-1:
LET x=x+1: GO TO 220
 320 IF r$="z" THEN  LET y=y-1:
LET x=x-1: GO TO 220
 330 IF r$=CHR$ 13 THEN  GU TO ■
00
 340 GO TO 230
 400 PRINT #1;AT 0,0;"
                      "
 410 PRINT #1;AT 0,25;"SPACE"
 420 PRINT #1;AT 0,5;"POKOLENIE"
 430 RANDOMIZE USR 23810
 440 PRINT #1;AT 0,0;PEEK 23800;
" ";
 450 IF INKEY$<>CHR$ 32 THEN  ■■
TO 430
 460 PRINT #1;AT 1,0;"x-CLS c-CO
NT z-ZMIANY "
 500 REM
 503 REM obsluga przerwania
 510 LET r$=INKEY$: IF r$="" THE
N  GO TO 510
 520 IF r$="x" THEN  GO TO 10
 530 IF r$="c" THEN  GO TO 400
 540 IF r$="z" THEN  PRINT #1;AT
 0,0;"
                      ";
FOR i=1 TO 80: NEXT i: GO TO 220
 550 ■■ TO 510
1000 REM
1003 REM ■■■■ 1
1010 LET end=10
1020 GO SUB 1900
1030 CLS
1050 LET end=250
1053 REM demo2
1060 FOR i=63 TO 115
1070 PLOT 102,i: ■■■W 52,0
1080 NEXT i
1090 GO SUB 1900: CLS
1100 REM
1103 REM demo 3
1110 LET end=100
1120 PLOT 50,88: ■■■■ 155,0
1130 GO SUB 1900
1140 CLS
1200 REM
1203 REM demo 4
1210 LET end=80
1220 PLOT 100,70: DRAW 50,0: DRA
W 0,30: DRAW -50,0: DRAW 0,-30:
■■■■ 50,30
1230 PLOT 150,70: DRAW -50,30
1240 GO SUB 1900
1250 RUN
1900 REM
1903 REM kolejne pokolenia-demo
1910 POKE 23800,0: PRINT #1;AT 0
,5;"Pokolenie        SPACE"
1920 FOR i=0 TO end
1930 RANDOMIZE USR 23810
1940 PRINT #1;AT 0,0;PEEK 23800;
"   ";
1950 IF INKEY$=CHR$ 32 THEN  GU
TO 1970
1960 NEXT i
1970 PRINT #1;AT 1,0;"KONIEC"
1980 FOR i=1 TO 200: NEXT i
1990 RETURN
2000 REM
2003 REM wczytanie kodu
2010 CLS : POKE 23756,0: POKE 23
757,112: POKE 23758,5
2020 FOR i=23760 TO 23760+111+25
6*4 STEP ■
2030 RANDOMIZE USR 3280
2040 PRINT AT 15,0;i;"  ";
2050 INPUT "  ";a$: IF LEN ■■ 1
6 THEN  ■■ TO 2050
2060 LET su=0
2070 FOR j=1 TO ■
2080 LET b$=a$(2*j-1 TO 2*j)
2090 GO SUB 2900
2100 LET su=su+s
2110 NEXT j
2120 PRINT a$;"  ";su
2130 INPUT "t-tak n-nie ";c$
2140 IF c$="t" THEN  GO TO 2200
2150 IF c$="n" THEN  ■■ TO 2030
2160 GO TO 2130
2200 FOR j=1 TO ■
2210 LET b$=a$(2*j-1 TO 2*j)
2220 GO SUB 2900
2230 POKE i+j-1,s
2240 NEXT j
2250 NEXT i
2260 PRINT ''"zapis wlacz magnet
ofon""nacisnij dowolny klawisz"
2270 SAVE "SUPERLIFE" LINE 1
2280 RUN
2900 REM
2903 REM ■■■■■■ kodu szesnastko
wego na dziesietny
2910 IF CODE b$(1)<58 THEN  LET
s=16*(CODE b$(1)-48): GO TO 2930
2920 IF CODE b$(1)>96 THEN  LET
s=16*(CODE ■■(1)-87)
2930 IF CODE b$(2)<58 THEN  LET
s=s+(CODE b$(2)-48): GO TO 2950
2940 IF CODE b$(2)>96 THEN  LET
s=s+(CODE ■■(2)-87)
2950 RETURN
```

Wydruk 2. ■■■ maszynowy programu SUPERLIFE

```
23760  1600032027535550   344
23768  45524c4946452720   510
23776  207f204a2e535a41   549
23784  4652414e454b10a0   615
23792  4242420042424242   462
23800  00424242eeda42f0   960
23808  da42cd1c5dcda260  1077
23816  cd3f5dcd4b5dcd43  1006
23824  60cd5960cd7e6021   946
23832  f85c34c90e0006b0   789
23840  11e0d105626b1120   709
23848  0019ebc5d5cdaa22  1079
23856  d1d5012000edb0d1  1077
23864  c13e008020e5c921   878
23872  00bb1100e9010016   460
23880  edb0c921fc5c5e23  1120
23888  56ebe52b7e234623   ■■■
23896  4eb0b111dfff1946  1021
23904  114000194eb0b1e1   762
23912  e5ca3060010000cb   779
23920  7e2b0104cb762b01   533
23928  0c2bcb4628010c11   398
23936  e0ff19cb4628010c   830
23944  23cb7e2b010ccb76   738
23952  2b010c113f0019cb   361
23960  4628010c23cb7e28   527
23968  010ccb762b010caf   562
23976  b0280d3e00b1fe02   628
23984  2b13fe0328f01805   400
23992  3e03912008e1e511   721
24000  001719cbfee1e501   960
24008  0000cb762b0104cb   569
24016  7e2b010ccb6e2b01   533
24024  0c11e0ff19cb6e28   ■■■
24032  010ccb762b010ccb   590
24040  7e2b010c11400019   ■■■
24048  cb6e28010ccb762b   727
24056  010ccb7e28010caf   570
24064  b0280d3e00b1fe02   628
24072  2b13fe0328f01805   400
24080  3e03912008e1e511   721
24088  001719cbf6e1e501   952
24096  0000cb6e2b0104cb   561
24104  762b010ccb662b01   517
24112  0c11e0ff19cb662b   878
24120  010ccb6e28010ccb   582
24128  762b010c11400019   277
24136  cb662b010ccb6e28   711
24144  010ccb762b010caf   562
24152  b0280d3e00b1fe02   ■■■
24160  2b13fe0328f01805   400
24168  3e03912008e1e511   721
24176  001719cbee1e501    944
24184  0000cb662b0104cb   ■■■
24192  6e2b010ccb5e2b01   501
24200  0c11e0ff19cb5e28   870
24208  010ccb662b010ccb   574
24216  6e28010c1140001⁹   26⁹
24224  cb5e28010ccb662b   695
24232  010ccb6e28010caf   ■■■
24240  b0280d3e00b1fe02   ■■■
24248  2b13fe0328f01805   400
24256  3e03912008e1e511   721
24264  001719cbe6e1e501   ⁹36
24272  0000cb5e28010cb    545
24280  5628010ccb662801   485
24288  0c11e0ff19cb5628   ■■■
24296  010ccb5e28010ccb   566
24304  6628010c1140001⁹   261

24312  cb5628010ccb5e28   67⁹
24320  010ccb662b010caf   546
24328  b0280d3e00b1fe02   ■■■
24336  2b13fe0328f01805   ■■■
24344  3e03912008e1e511   721
24352  001719cbd6e1e501   ⁹28
24360  0000cb5628010cb    ■■■
24368  4e28010ccb5e2801   469
24376  0c11e0ff19cb4e28   854
24384  010ccb5628010ccb   558
24392  5e28010c11400019   ■■■
24400  cb4e28010ccb5628   663
24408  010ccb5e28010caf   ■■■
24416  b0280d3e00b1fe02   ■■■
24424  2b13fe0328f01805   400
24432  3e03912008e1e511   721
24440  001719cbd6e1e501   ⁹20
24448  0000cb4e28010cb    52⁹
24456  4628010ccb562801   453
24464  0c11e0ff19cb4628   846
24472  010ccb4e28010ccb   ■■■
24480  5628010c11400019   245
24488  cb4628010ccb4e28   647
24496  010ccb5628010caf   530
24504  b0280d3e00b1fe02   628
24512  2b13fe0328f01805   400
24520  3e03912008e1e511   721
24528  001719cbcee1e501   912
24536  0000cb46280104cb   521
24544  4e28010c23cb7e28   535
24552  010c11e0ff19cb7e   863
24560  28010c2bcb462801   410
24568  0ccb4e28010c1140   427
24576  0019cb4628010ccb   554
24584  4e28010c23cb7e28   535
24592  010caf80280d3e00   431
24600  81fe022813fe0328   741
24608  0f18053e03912008   294
24616  e1e51100171cbc6    920
24624  21ff5c5e2356e123   855
24632  7aacc2525d7badc2  1153
24640  525dc92100d20100   620
24648  e9c5e50a5eaa77e1   126⁹
24656  c103233ee8942041   ⁹46
24664  c92100bb1100d201   64⁹
24672  2000edb011e0e701   918
24680  2000edb021ffd1cb  1145
24688  8623cbbe111f001⁹   635
24696  3ee8942042c9e00    ⁹31
24704  06b011e0d105626b   842
24712  1120001⁹ebc5d5cd   ⁹24
24720  aa22d1d5012000eb   894
24728  edb0d1c13e00b020  1037
24736  e4c92120d2e53e00   995
24744  862011e1233ee8⁹4   885
24752  20f321fd5c36d22b   960
24760  36001b0d01fd5ce1   662
24768  11dfff1⁹7c020b7d   782
24776  0221e0e7e53e0086   915
24784  2010e12b3ed19420   767
24792  f321005d36e82b36   752
24800  00c9112100e11901   502
24808  005d7c027d0b02c9   ■■■
24816  ff1946114000194e   534
24824  b0b1e1e5ca226001  1140
24832  0000cb7e2b0104cb   577
24840  7628010c2bcb4e28   527
24848  010c11e0ff19cb46   807
24856  28010c23cb7e2801   458
24864  0ccb7628010c113f   466
24872  9019cb4628010c23   386
24880  cb7e2b010ccb7628   743
24888  010caf802b0d006e   47⁹
```

TURBOTAPE 64

Problem dużej czasochłonności zapisu i odczytu programów komputerowych na taśmie magnetofonowej gnębi zapewne nie jednego posiadacza mikrokomputera. Najlepszym (technicznie) rozwiązaniem jest przejście na współpracę ze stacją dysków elastycznych. W większości wypadków możliwość zakupu stacji dyskietek przez przeciętnego programistę-amatora jest zupełnie nierealna z powodu jej ceny (ok. 250 tys. zł). Na dodatek nie ma chętnych, by sprzedać to urządzenie. Pozostaje więc jedynie poszukiwanie dogodnych rozwiązań programowych.

Użytkownicy mikrokomputerów Commodore 64 mogą częściowo rozwiązać ten problem korzystając z programu **TURBOTAPE 64**. Program ten, napisany przez Stephana Seinza z RFN, prawie dziesięciokrotnie przyspiesza wykonanie operacji zapisu, odczytu i weryfikacji (realizowane rozkazami *LOAD, SAVE* i *VERIFY*). Należy zaznaczyć, że zapis i odczyt zbiorów logicznych (realizowane instrukcjami *OPEN, PRINT#, GET#, INPUT#, CLOSE*) nie są przyspieszane.

Przedstawione dalej rozwiązanie dość znacznie różni się od oryginalnej wersji programu TURBOTAPE 64, sprzedawanej na wolnym rynku po ok. 3000 zł. Zrezygnowano w nim z części opisowej, wyświetlanej na ekranie jako instrukcja obsługi. Ponadto zrekonfigurowano pamięć komputera w taki sposób, aby program był ładowany bezpośrednio do obszaru pamięci, który będzie zajmował. Dzięki temu czas załadowywania TURBOTAPE 64 skrócił się prawie o połowę.

Wprowadzanie programu należy przeprowadzić w dwóch fazach. W pierwszej, należy wpisać program przedstawiony na wydruku i uruchomić go poleceniem *RUN*. W ten sposób kod maszynowy zostanie umieszczony w obszarze pamięci przeznaczonym dla naszego programu. W drugiej, należy tak zrekonfigurować pamięć, aby wskaźniki początku i końca obszaru pamięci przeznaczonego dla języka BASIC wskazywały dokładnie na początek i koniec obszaru pamięci, w którym jest zapisany program TURBOTAPE 64 (adresy od 50000 do 50774, dziesiętnie). Rekonfigurację tę należy przeprowadzić wpisując w trybie bezpośrednim:

```
POKE 43,80 : POKE 44,195 : REM POCZATEK OD
ADRESU 50000 POKE 45,87 : POKE 46,198
: NEW
REM KONIEC NA ADRESIE 50774+1
```

Po przeprowadzeniu rekonfiguracji program jest gotowy do zapisania na kasecie magnetofonowej. Zapis należy zrealizować używając polecenia:

```
SAVE " TURBOTAPE 64", 1,1
```

Takie użycie rozkazu SAVE spowoduje zapisanie (w nagłówku programu) informacji, do jakiego obszaru pamięci będzie wpisywany program (w tym wypadku do obszaru 50000 ... 50774).

Zapisany i gotowy do użytku program uruchamia się poleceniem *SYS 50000*. Odpowiedniki poleceń zapisu, odczytu i weryfikacji podano w tabeli. Należy zwrócić uwagę, że program użytkownika musi być uprzednio zapisany w systemie TURBO, aby mógł być odczytany z taśmy w tym systemie. Standardowo zapisanych programów nie można odczytywać programem TURBOTAPE 64, tak samo jak programów zapisanych przez TURBOTAPE 64 nie da się odczytać wykorzystując standardowe polecenia.

Procedura odczytu i weryfikacji w systemie TURBO różni się nieco od standardowej Po odczytaniu nagłówka i wyświetleniu tytułu programu, operator, aby kontynuować dalszy odczyt, musi wcisnąć klawisz *SPACE* (odstęp) lub *Commodore LOGO KEY* i *RETURN*. W trybie standardowym komputer sam przechodzi do dalszego odczytywania po upływie określonego czasu.

Taśma przygotowana do zapisu programu w systemie TURBO powinna być ustawiona za rozbiegówką ze względu na znaczne skrócenie czasu, w którym podawany jest sygnał poprzedzający nagłówek. Odstęp po zapisie programu (oddzielający ewentualne dalsze nagrania) można zrealizować za pomocą instrukcji *POKE* 192,0 – spowoduje to uruchomienie silnika magnetofonu, chociaż żaden sygnał nie będzie nagrywany. Po zrobieniu odpowiedniego odstępu należy wyłączyć magnetofon za pomocą klawisza *STOP*.

Zapis bądź odczyt można przerwać wciskając klawisz *RUN/STOP*. Informacja o przerwaniu nie ukazuje się na ekranie i dlatego w celu przywrocenia standardowego trybu pracy komputera należy wcisnąć jednocześnie klawisze *RUN/STOP* i *RESTORE*.

Nia zaleca się zapisywania na tej samej taśmie programów nagranych w systemie TURBO i normalnie. W wypadku gdy komputer poszukuje programu nagranego w trybie standardowym, a natrafi na początek programu w systemie TURBO, wówczas najczęściej następuje przerwanie odczytu programu lub komputer blokuje się.

Niestety nie wszystkie programy można zapisywać i później odczytywać w systemie TURBO. Dotyczy to przede wszystkim tych programów, które wykorzystują dodatkowe 4 KB pamięci RAM (o adresach od 49152 do 53247 dziesiętnie). Zwykle próba uruchomienia takiego programu powoduje częściowe skasowanie systemu TURBO i powoduje natychmiastowe zablokowanie się komputera.

Program TURBOTAPE 64 zajmuje dodatkowe 4 KB pamięci RAM, niedostępnej w standardowym trybie pracy dla programów w języku BASIC, ale za to doskonale nadającej się do przechowywania danych maszynowych. Obszar ten można skasować praktycznie tylko trzema sposobami:

– przez wpisanie pętli kasującej (FOR X=49152 TO 53247 : POKE x,0)
– przez wpisanie innego programu, który w całości skasuje poprzedni program
– przez wyłączenie komputera z sieci.

Ani instrukcja *NEW*, ani nawet tzw. procedura „zimnego startu", realizowana instrukcją *SYS 64738* (rekonfiguruje komputer do stanu analogicznego jak po włączeniu zasilania) nie są w stanie skasować programu TURBOTAPE 64. Można więc uruchomić go ponownie, nawet po skasowaniu zawartości pamięci RAM; wystarczy napisać **SYS 50000** i wcisnąć **RETURN**...

Praktyka dowiodła, że najlepszym sposobem przechowywania TURBOTAPE jest nagrywanie go na oddzielnej kasecie po obu jej stronach. Staje się on wtedy łatwo dostępny, np. przed załadowaniem nowej gry lub w wypadku zablokowania się komputera. Nagrywanie takiej kasety należy przeprowadzić zaraz po wpisaniu do komputera programu TURBO po raz pierwszy i odpowiednim zrekonfigurowaniu pamięci (tak jak to zostało opisane). Zapis taki można powielić, wpisując w trybie bezpośrednim następujący krótki program:

```
FOR J = 1 TO 20 : SAVE „ TURBOTAPE 64
"1,1
NEXT J
lub
FOR J = 1 TO 20 : SAVE „ TURBOTAPE 64
",1,1
POKE 192,0
FOR K = 1 TO 1000 : NEXT K : NEXT J
```

Druga wersja pozwoli umieścić na taśmie odstępy między programami: długość odstępu zależy od maksymalnej wartości zmiennej K.

KLAUDIUSZ DYBOWSKI
MICHAŁ SILSKI

TURBO TAPE 64 V.ML

```
10 REM ■■■ TURBO TAPE 64 V.ML ■■■
20 :
30 FORX=50000TO50774:READA:POKEX,A:CK=CK+A:NEXT
40 IFCK<>93316THENPRINT"POMYLKA W DANYCH !":PRINTCK:STOP
50 PRINTCHR$(147);"SYS 50000 TO START":NEW
60 :
70 DATA 169,091,141,008,003,169,195,141,009,003,096
80 DATA 032,115,000,240,004,201,095,240,003,076,231
90 DATA 167,032,115,000,201,083,240,011,201,076,240
100 DATA 016,201,086,240,021,076,008,175,032,115,000
110 DATA 032,240,195,076,174,167,032,115,000,032,224
120 DATA 196,076,174,167,032,115,000,032,227,196,076
130 DATA 174,167,000,000,000,000,000,000,000,000,000
140 DATA 000,000,000,000,000,000,000,000,000,000,000
150 DATA 000,000,000,000,000,000,000,000,000,000,000
160 DATA 000,000,000,000,000,000,000,000,000,000,000
170 DATA 000,000,000,000,000,000,000,000,000,000,000
180 DATA 000,000,000,000,000,000,000,000,000,000,000
190 DATA 000,000,000,000,000,000,000,000,000,000,000
200 DATA 000,000,000,000,000,000,000,000,000,000,000
210 DATA 000,000,000,000,000,000,162,005,134,171,032
220 DATA 212,225,162,004,181,042,149,171,202,208,249
230 DATA 032,056,248,032,143,246,032,125,196,032,145
240 DATA 196,165,185,024,105,001,202,032,177,196,162
250 DATA 008,185,172,000,032,177,196,162,006,200,192
260 DATA 005,234,208,242,160,000,162,004,177,187,196
270 DATA 183,144,003,169,032,202,032,177,196,162,005
280 DATA 200,192,187,208,237,169,002,133,171,032,145
290 DATA 196,152,032,177,196,132,215,162,007,234,177
300 DATA 172,032,177,196,162,003,230,172,208,004,230
310 DATA 173,202,202,165,172,197,174,165,173,229,175
320 DATA 144,231,234,165,215,032,177,196,162,007,136
330 DATA 208,246,200,132,192,088,024,169,000,141,160
340 DATA 002,076,147,252,160,000,132,192,173,017,208
350 DATA 041,239,141,017,208,202,208,253,136,208,250
360 DATA 120,096,160,000,169,002,032,177,196,162,007
370 DATA 136,192,009,208,244,162,005,198,171,208,238
380 DATA 152,032,177,196,162,007,136,208,247,202,202
390 DATA 096,133,189,069,215,133,215,169,008,133,163
400 DATA 006,189,165,001,041,247,032,211,196,162,017
410 DATA 234,009,008,032,211,196,162,014,198,163,208
420 DATA 233,096,202,208,253,144,005,162,011,202,208
430 DATA 253,133,001,096,162,000,044,162,001,164,043
440 DATA 165,044,134,010,134,147,132,195,133,196,032
450 DATA 212,225,032,253,196,032,122,225,076,116,164
460 DATA 032,097,197,165,171,201,002,240,008,201,001
470 DATA 208,243,165,185,240,010,173,060,003,133,195
480 DATA 173,061,003,133,196,032,080,247,032,228,255
490 DATA 240,251,032,044,168,164,183,240,011,136,177
500 DATA 187,217,065,003,208,206,152,208,245,132,144
510 DATA 032,210,245,173,062,003,056,237,060,003,008
520 DATA 024,101,195,133,174,173,063,003,101,196,040
530 DATA 237,061,003,133,175,032,118,197,165,189,069
540 DATA 215,005,144,240,004,169,255,133,144,076,169
550 DATA 245,032,175,197,201,000,240,249,133,171,032
560 DATA 221,197,145,178,200,192,192,208,246,240,045
570 DATA 032,175,197,032,221,197,196,147,208,002,145
580 DATA 195,209,195,240,002,134,144,069,215,133,215
590 DATA 230,195,208,002,230,196,165,195,197,174,165
600 DATA 196,229,175,144,221,032,221,197,032,125,196
610 DATA 200,132,192,088,024,169,000,141,160,002,076
620 DATA 147,252,032,023,248,032,125,196,132,215,169
630 DATA 007,141,006,221,162,001,032,240,197,038,189
640 DATA 165,189,201,002,208,245,160,009,032,221,197
650 DATA 201,002,240,249,196,189,208,232,032,221,197
660 DATA 136,208,246,096,169,008,133,163,032,240,197
670 DATA 038,189,234,234,234,198,163,208,244,165,189
680 DATA 096,169,016,044,013,220,240,251,173,013,221
690 DATA 142,007,221,072,169,025,141,015,221,104,074
700 DATA 074,096,000,000,000,000,000,000,000,000,000
710 DATA 197,038,189,165,189,201,002,208,245,160,009
720 DATA 032,221,197,201,002,240,249,196,189,208,232
730 DATA 032,221,197,136,208,246,096,169,008,133,163
740 DATA 032,240,197,038,189,234,234,234,198,163,208
750 DATA 244,165,189,096,169,016,044,013,220,240,251
760 DATA 173,013,221,142,007,221,072,169,025,141,015
770 DATA 221,104,074,074,096

READY.
```

Nowe kostki

Badania nad zastosowaniem GaAs (arsenku galu) w układach scalonych przynoszą już wymierne efekty. Firma LEP opracowała pamięć statyczną RAM, opartą na technologii DCFL (ang. Direct Coupled Fet Logic), o organizacji 1K × 1. Od produkowanej przez CEMI kostki 2102, różni się ona jednak zasadniczo, ponieważ jej czas dostępu (okres, jaki musi upłynąć od ustawienia linii sterujących do odczytu – zapisu danej) wynosi, w zależności od wersji, 2 bądź 3 nanosekundy. Układ zawiera 4979 tranzystorów i 2324 oporniki umieszczone na powierzchni 2,5 × 2,5 mm. Pobór mocy – 90 mW. Sądzi się, że całość produkcji tych układów zostanie przeznaczona do budowy superkomputerów.

★ ★ ★

Rynek układów scalonych VLSI musi być bardzo chłonny, skoro kolejny gigant informatyczny, AT&T, postanowił zaznaczyć na nim swoją obecność. Start nastąpił od razu z najwyższego poziomu: pierwszymi kostkami są układy tworzące nową rodzinę mikroprocesora 32-bitowego drugiej generacji – WE 32100. Układy, z których większość jest już produkowana seryjnie, wytwarzane są w technologii CMOS 1,5 µm i pracują z zegarem o częstotliwości 14 MHz. AT&T wyprzedziła tym samym swoich konkurentów – pracujących nad „32-bitowcami" CMOS 1,5 µm – o 12 do 18 miesięcy.

Jesienią zeszłego roku na rynku były już dostępne (według producenta – w nieograniczonych ilościach) następujące układy: jednostka centralna CPU WE 32100, układ zarządzania pamięcią MMU WE 32101, zegar systemowy WE 32102, interfejs systemowy WE 32105.

Na koniec 1985 roku zapowiedziano: układ wspomagania operacji zmiennoprzecinkowych MAU i sterownik pamięci dynamicznych DRC WE 32103.

Rodzina WE 32100 wyposażona została już w system operacyjny – oczywiście UNIX V, bo UNIX to przecież twór właśnie AT&T.

Mikroprocesor 32100 ma 32-bitową szynę adresową. Umieszczany jest w obudowie „grid pin array" („pole siatki szpilek") o 132 wyprowadzeniach. Na marginesie, na Zachodzie przyjęło się już nazywać takie obudowy mianem „fakir".

Ceny układów – jeszcze nie na naszą kieszeń: pomiędzy 450 a 675 dolarów za kostkę.

★ ★ ★

Firma Fujitsu oferuje ostatnio pamięci statyczne RAM o nietypowej pojemności: 72 KB. Układ, oznaczony MB 81C79, ma organizację 8 K × 9 bitów (bity parzystości – patrz artykuł na str.7). Czas dostępu: max. 45 ns; pobór mocy: 83 mW w stanie pasywnym i 665 mW w czasie wykonywania operacji. Obudowa DIL z 28 wyprowadzeniami. Układ może znaleźć zastosowanie w małych systemach, w których niezawodność jest parametrem krytycznym (np. sprzęt medyczny).

ATARI 800 XL

Coraz więcej komputerów domowych trafia do naszych mieszkań. Ostatnio (artykuł pisany w listopadzie 1985) również za pośrednictwem Pewexu. Pierwsza, informacyjna partia ATARI 800 XL rozeszła się błyskawicznie.

Poniższy, wprowadzający materiał o ATARI 800 XL jest dedykowany tym, których dylemat nie polega na tym czy kupić, lecz co kupić (w wypadku ATARI za dolary, niestety). Nie poruszamy w nim zagadnień sprzętowych. Będą się one pojawiać wraz z ewentualnym upowszechnieniem tego komputera.

Liczymy na współpracę Czytelników. Przypominamy: jeżeli oczekujesz artykułów poświęconych Twojemu komputerowi – sam również podziel się posiadanymi informacjami z innymi Czytelnikami.

Kalifornijska firma ATARI mieści się w samej Dolinie Krzemowej. Nie może zatem narzekać na luki w dopływie technologii. Na brak funduszy chyba też, jest bowiem częścią składową Warrner Brothers Communications Company – giganta znanego w Polsce głównie z czołówek filmów. Wszystko to razem sprawia, że produkty ATARI są konkurencyjne. Popularność różnych modeli komputerów ATARI dorównywała w swoim czasie APPLE II i II+; obecnie ATARI 520 konkuruje z Macintoshem.

Ogólnie można powiedzieć, że ATARI 800 XL został skonstruowany dla potrzeb domu. Dzieci z pewnością oczaruje bajecznie kolorowy i doskonale animowany obraz; Pani Domu może sobie zaplanować budżet domowy, wreszcie Pan Domu może uruchomić program, korzystając z Microsoft BASIC lub np. założyc na dyskietce – niedostępny dla osób postronnych – zbiór telefonów koleżanek z pracy, albo zagrać z komputerem w pokera. A wszystko to za cenę niższą (na Zachodzie) niż cena telewizora kolorowego.

Wyglądem zewnętrznym ATARI 800 XL przypomina trochę Commodore 64. W pełni profesjonalna, nie budząca od strony mechanicznej żadnych zastrzeżeń, klawiatura typu QWERTY jest opisana przejrzyście i bez zbędnego przegadania, charakterystycznego np. dla ZX SPECTRUM. Klawiatura zawiera również klawisze, które można określić mianem systemowych, należą do nich RESET, (którego brak odczuwają posiadacze wspomnianego już Spectrum) oraz bardzo użyteczne przy grach: START, OPTION, SELECT. Estetycznie wkomponowana dioda LED informuje, że komputer jest włączony (przełącznik umieszczony z tyłu obudowy).
Zasilanie systemu odbywa się poprzez zewnętrzny zasilacz. Nie sprawia on żadnych kłopotów: w czasie, gdy powstał ten artykuł, pracował bez przerwy przez 12 godzin, nie nagrzewając się specjalnie.

Tuż obok gniazda zasilającego system umieszczone jest wyjście sygnału telewizyjnego. Komputer generuje sygnał antenowy w systemie PAL. Zmartwi to zapewne tych potencjalnych uzytkowników, którzy dysponują telewizorami w systemie SECAM. Jest to jednak cecha większosci komputerow domowych. Są one przecież produkowane z myślą o rynkach, na których niepodzielnie królują PAL i NTSC ... Pewnym ułatwieniem (dla wprawnych majsterkowiczów) jest dostępność na zewnątrz komputera sygnałów RGBY do sterowania monitorem kolorowym.

Posiadaczy telewizorów czarno-białych kłopoty te nie dotyczą. Napotkają oni jednak następujący problem: komputer generuje dźwięki poprzez tor foniczny odbiornika telewizyjnego, a sygnał ten nie jest dobrze dekodowany przez odbiorniki monochromatyczne. Można jednak dołączyć do ATARI wzmacniacz akustyczny, korzystając z wyjścia na gnieździe monitorowym.

ATARI 800 XL nie ma wyprowadzonej na zewnątrz magistrali systemowej. Dostępne są jednak liczne porty umożliwiające sterowanie właściwie wszystkimi typowymi urządzeniami peryferyjnymi: drukarką, złączem szeregowym RS-232C, modemem (to propozycja dla tych, którzy nie stracili jeszcze nadziei na rozwój w Polsce ogólnodostępnej sieci komputerowej), manipulatorami (ang. joystick) oraz stacjami napędów dyskietek 5,25-calowych.

Ta ostatnia możliwość stawia ATARI 800 XL w rzędzie sprzętu wyrastającego ponad poziom gier i prostych programów obliczeniowych. Tym bardziej że kolejne stacje pamięci dyskowej mogą być dołączone do komputera w sposób natychmiastowy, czyniąc zeń system o pokaźnej mocy przetwarzania danych.

Centralnym elementem konstrukcji jest mikroprocesor 6502, współpracujący z pamięcią RAM o pojemności 48 KB i pamięcią ROM o pojemności 16 KB. Upodobnia to ATARI 800 XL do takich komputerów jak Apple II lub Commodore. Proszę nie liczyć jednak na to, że oznacza to przenośność oprogramowania. Wprawdzie procesor taki sam – ale cała reszta inna i aby przystosować do ATARI programy dobrze znane z Appie II, trzeba zejsc na poziom sprzętu i asemblera. A to już zadanie dla profesjonalistów – programistów systemowych. Ci wywiązali się zresztą z zadania, tworząc bogatą bibliotekę oprogramowania. Są to w większości właśnie adaptacje dobrze znanych programów użytkowych. Wersje te nie są w Polsce rozpowszechnione, ale – jak uczy doświadczenie – to tylko kwestia czasu. Zwykle bowiem ilość dostępnego (z różnych źródeł) oprogramowania jest wprost proporcjonalna do liczby użytkowanych komputerów danego typu.

Zawarty w pamięci stałej systemu interpreter języka BASIC należy z pewnością do silnych punktów komputera ATARI 800 XL. Jest on ogólniejszym i pełniejszym narzędziem niż interpreter popularnego ZX Spectrum. Wynika to przede wszystkim z możliwości sterowania pełniejszym zestawem urządzeń wejścia-wyjścia i pamięci masowei.

Omówimy tu pokrótce funkcje i polecenia języka ATARI BASIC, uwypuklając rozszerzenia w stosunku do innych rozwiązań.

ATARI BASIC oprócz typowych instrukcji **LET, POKE, PEEK, DIM, DATA, READ, RESTORE I REM** zawiera użyteczne polecenie **CLEAR** pozwalające na wyczyszczenie zawartości pamięci zarezerwowanej poleceniem DIM.

Wśród instrukcji sterujących występują – oprócz instrukcji prostego skoku **GOTO**, prostego przejścia do podprogramu **GOSUB** i instrukcji **IF...THEN** – takie konstrukcje jak **ON...GOTO; ON...GOSUB**. Ciekawa jest możliwość przerwania wykonywania pętli **FOR...NEXT** instrukcją **POP**, która oczyszcza stos zakładany na czas wykonywania pętli i w konsekwencji pozwala na opuszczenie jej w dowolnej chwili. Wykonywanie programu w języku BASIC może być śledzone dzięki instrukcji **TRAP** i zawieszane instrukcją **STOP**.

Bogate są możliwości graficzne interpretera. Przy użyciu instrukcji **GRAPHICS** można zaprogramować 16 sposobów prezentacji wyników: 3 tekstowe i 13 graficznych. Największa rozdzielczość dostępna na ekranie to 320×192, a największą liczbę kolorów oferuje tryb GRAPHICS 11:80×192 punkty, każdy w jednym z szesnastu możliwych kolorów.

Do tworzenia grafiki służą instrukcje **SET, COLOR, PLOT, DROWTO** – ta ostatnia realizuje kreślenie linii od aktualnego położenia wskaznika do określonego punktu. Wskaźnik może być biernie przenoszony instrukcją **POSITION**, a instrukcja **LOCATE** odczytuje kod koloru w punkcie aktualnie wskazywanym.

Do operacji na łańcuchach (napisach) służą **ASC** (ii), **CHR$**, **LEN**, **STR$** oraz kilka innych funkcji umożliwiających operowanie na fragmentach łańcuchów. Istnieje również możliwość łączenia napisów.

Zwykle komputery domowe nie są wyposażone w osobny program typu MONITOR, który pozwala na bezpośrednie korzystanie z tzw. zasobów sprzętowych maszyny, czyli mówiąc po prostu, z możliwości poszczegól-

nych układów scalonych. Część z tych możliwości przerzuca się do interpretera języka wysokiego poziomu. Konstruktorzy ATARI 800 XL postąpili podobnie.

Obok wymienionych powyżej instrukcji **PEEK** i **POKE** mamy do dyspozycji funkcję **ADR**, przypisującą wyspecyfikowanej zmiennej adres początkowy w pamięci RAM interesującego nas ciągu znaków (łańcucha), oraz funkcję **USR** – wywołującą podprogram w kodzie maszynowym. Dla tych, którzy stosowali już tego typu zabiegi na innych mikrokomputerach, dodajmy, że funkcja ta umożliwia przekazywanie dowolnej liczby parametrów, dokonuje się to bowiem nie za pośrednictwem rejestrów mikroprocesora, lecz przez stos.

Jednym z głównych zadań programów w języku BASIC jest wykonywanie obliczeń. Przydatny do tego celu jest zestaw funkcji arytmetycznych, w skład którego wchodzą funkcje **LOG, CLOG, EXP, INT, ABS, RND, SGN, SQR, SIN, COS, TNG, ATN.** Argumenty funkcji trygonometrycznych są traktowane jako zapisane w stopniach, jeżeli użycie funkcji poprzedzi instrukcja **DEG,** lub w radianach, jeżeli w czasie wykonywania programu zostanie napotkane polecenie obliczeń w radianach – **RAD**.

Cechą wartą szczególnego podkreślenia jest bogactwo możliwości sterowania przepływem informacji, zarówno do, jak i z komputera. W przeciwieństwie do większości innych konstrukcji, ATARI BASIC operuje pojęciem znanym w większych maszyn: strumieniami we-wy. Przez strumień należy rozumieć umowny kanał dla przepływu informacji między dwoma urządzeniami. Strumień taki może być otwarty bądź zamknięty, podobnie jak zwykły kanał. Kanałami takimi można łączyć komputer z siedmioma dostępnymi w systemie urządzeniami peryferyjnymi: klawiaturą, drukarką, magnetofonem kasetowym, ekranem telewizora, konsolą (zewnętrznym urządzeniem we-wy – np. terminalem typu DZM-180 KSR), urządzeniem pracującym poprzez łącze szeregowe w standardzie RS-232C i stacjami dyskietek (do 4 w systemie). Kanał jest otwierany poleceniem **OPEN**, a zamykany poleceniem **CLOSE.** Polecenia te mają szereg parametrów i opcji, których dokładny opis znajduje się w podręczniku firmowym. Aby zasygnalizować możliwości takiego rozwiązania, odwołajmy się do przykładu.

Przyjmijmy, że przez kanał nr 3 będziemy przesyłać informację do – za stacji dysków nr 2. Informacje te mają być pobierane lub zapisywane do zbioru o nazwie OLA, składającego się z danych binarnych. Dokonamy tego jedną instrukcją: **OPEN 3,12,0, „D2:OLA.BIN"**.

Na dostępnych, dzięki uprzednio otwartym kanałom, zbiorach można wykonywać operacje **INPUT, OUTPUT** oraz **GET** i **PUT** (**GET** i **PUT** dotyczą pojedynczych bajtów, a można je wykonać nawet na zbiorach dyskowych). Możemy ustawić logicznie głowicę dysku nad zadanym bajtem w pewnym zbiorze **(NOTE** i **POINT)** oraz odczytywać stan urządzeń obsługujących otwarte kanały. Automatycznie są otwierane strumienie: wejściowy z klawiatury i wyjściowy na monitor TV. Również dostęp do drukarki (instrukcją **LPRINT)** nie wymaga uprzedniego otwierania.

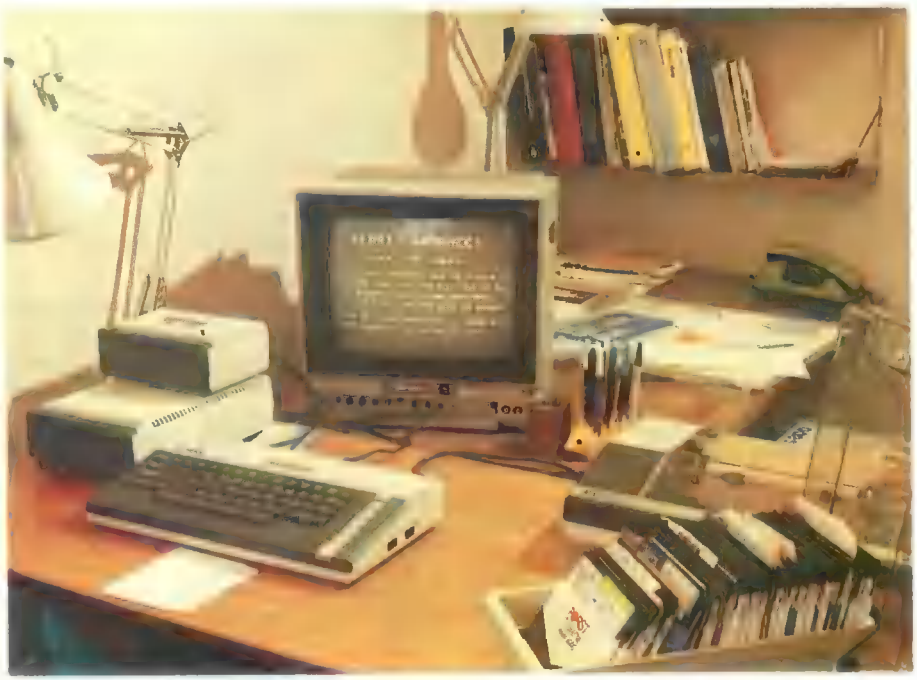
Zwolennicy gier, którzy nie muszą korzystać z wyrafinowanych konfiguracji systemu, mają do dyspozycji użyteczne instrukcje sprawdzania stanu manipulatorów oraz generowania dźwięków. Instrukcja **SOUND** ma 4 parametry, określające: numer kanału dźwiękowego, czas trwania dźwięku oraz jego wysokość i siłę.

Nasze programy (w tym gry) mogą być zapisywane na taśmie lub na innych urządzeniach wyjściowych (w tym na dysku), po czym powtórnie ładowane do pamięci. Wprost z interpretera języka BASIC posiadacze napędu dyskietek mogą przejść do systemu operacyjnego DOS (ang. Disk Operating System).

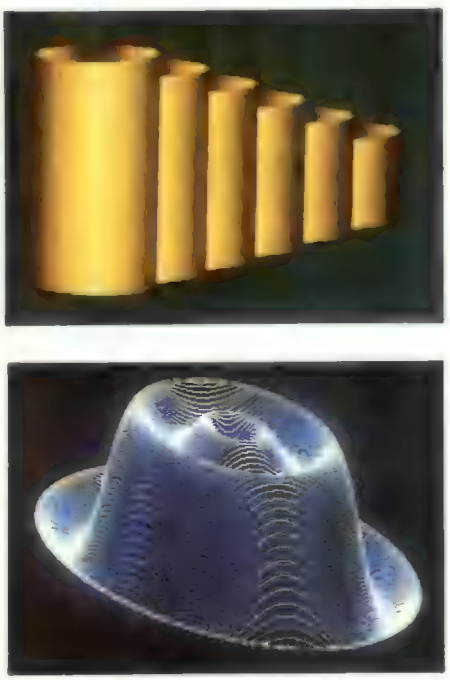
DOS jest zapisanym na dyskietce programem umożliwiającym manipulację zbiorami, zapisanymi również na dyskietce. Przez owo „manipulowanie" należy rozumieć takie operacje, jak: otwieranie nowego zbioru, kopiowanie i zmiana jego nazwy, ochrona zbioru przed dostępem przez osoby nieupoważnione, kopiowanie i formatowanie całych dyskietek itd. Zawartość każdej dyskietki może być prezentowana na życzenie. Umieszczony na dyskietce plik może być, obok nazwy, zaopatrzony w dodatkowy przyrostek określający, z jakiego typu zbiorem mamy do czynienia. Programy zapisane na dysku mogą być ładowane do pamięci i wykonywane.

ATARI-DOS ma właściwie wszystkie typowe możliwości systemu operacyjnego. W swoich możliwościach i konstrukcji jest podobny do systemu operacyjnego Apple II.

Inną ciekawą cechą ATARI 800 XL jest możliwość realizacji programów zawartych w tzw. cartige'u. Jest to specjalnie opakowana pamięć stała zawierająca program – najczęściej grę. Program ten, dzięki odpowiedniej konstrukcji obudowy kostki ROM, wy-



Fot. 1. Wymarzone stanowisko pracy: ATARI 800 XL, stacja dysków, monitor kolorowy o wysokiej rozdzielczości, drukarka, dyskietki ...

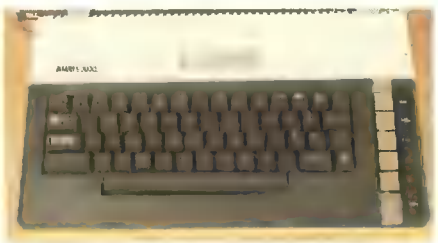


Fot. 2, 3. Przykłady możliwości graficznych

Fot. 4. Urządzenia zewnętrzne dołączamy poprzez gniazda umieszczone ściankach komputera: ściance bocznej – od lewej – złącza dla manipulatorów; tyłu – dla stacji dysków; dalej gniazda i monitora TV; prawej strony włącznik napięcia

mienia się pamięci komputera interpreterem języka BASIC, dzięki czemu nie zostaje ograniczony obszar dostępnej pamięci RAM. Wadą takiego rozwiązania jest brak możliwości skopiowania zawartości cartige'u, np. na taśmę. Należy jednak sądzić, że i ta trudność zostanie z czasem przełamana (opis podobnych zabiegów odbezpieczania programów dla ZX Spectrum można znaleźć artykule „Tajemnica programowych piratów" Informatyka 3,1985).

Programy dla ATARI są oferowane postaci pamięci stałych, kaset magnetofonowych lub dyskietek. Są wśród nich bazy danych, programy kalkulacyjne, procesory tekstów, kompilatory i interpretery języków algorytmicznych (np. LOGO lub standardowego Microsoft BASIC). Większość jednak użytkowników, przynajmniej na początku, wybierze wbudowany BASIC. Warto więc pokrótce przedstawić wyniki testów szybkości jego działania. Wykorzystano testy przedstawione na stronie 20 numerze 3,1984 Informatyki. Oto ich wyniki, kolejności programów: (w sekundach) 2,2; 7,4; 19,6; 22,9; 26,3; 40,2; 60,7. Suma czasów wykonywania testów wyniosła 179,3 s, co stawia ATARI 800 XL za np. Apple II (134,9 s), ale przed ZX Spectrum (205,4 s). Zwraca również uwagę doskonały generator liczb losowych, co ogromne znaczenie wielu zastosowaniach.



Fot. 5. klawiatury i przyciski funkcyjne

Reasumując, ATARI 800 XL może być cennym nabytkiem dla każdego domu, jest bowiem komputerem budowanym myślą tego typu potrzebach. Uwaga domowym charakterze opisanego mikrokomputera skierowana jest do tych wszystkich, którzy chcą zastosować ATARI 800 XL na przykład do zarządzania fabryką lub do prowadzenia intensywnego nadzoru stanu zdrowia pacjentów. We właściwej, domowej skali komputer ten długo będzie bawił, uczył i pomagał rozwiązywaniu nawet złożonych zagadnień.

JERZY ORKISZEWSKI

CZYM JEST UNIX (2)

POLECENIA SYSTEMOWE

Kontynuujemy cykl artykułów o systemie operacyjnym UNIX, który staje się standardem dla mikrokomputerów profesjonalnych.

Wykonywanie operacji na plikach jest jednym z głównych zadań systemu operacyjnego. Zgodnie z podziałem plików na skorowidze, pliki zwykłe i pliki specjalne, omówienie poleceń podzielono w ten sam sposób, przedstawiając najważniejsze z nich i najbardziej użyteczne dla początkującego użytkownika.

Ponieważ polecenia można wydawać wyłącznie po zgłoszeniu gotowości przez system operacyjny znakiem $ (w niektórych systemach jest to znak %), w poniższych przykładach konsekwentnie przestrzegano tej zasady. Ponadto założono, że bezpośrednio po zarejestrowaniu się (ang. logging in) użytkownika w systemie UNIX, następuje przejście do jego skorowidza macierzystego (ang. home directory), w prezentowanych przykładach zal.

POLECENIA DOTYCZĄCE SKOROWIDZÓW

Poniżej omówiono polecenia odnoszące się wyłącznie do skorowidzów. Polecenia dotyczące skorowidzów, lecz odnoszące się także do plikow zwykłych, przedstawiono w następnym punkcie.

mkdir

Polecenie **mkdir** (ang. make a directory) dotyczy utworzenia skorowidza. Przykładowo, wymieniony uprzednio skorowidz *zal* należy utworzyć poleceniem:

$mkdir *zal*

Podobnie, pozostałe skorowidze w skorowidzu *usr* tworzy się poleceniem:

$mkdir *koss schw*

$

Przy tworzeniu skorowidza automatycznie tworzone są dwa jego elementy standardowe, oznaczające skorowidz tworzony (.) i skorowidz nadrzędny (..).

Ściśle mówiąc, można utworzyć jednocześnie wszystkie skorowidze użytkowników z rys. 2 na str. 21 pierwszego zeszytu „Mikroklanu", wydając polecenie[1]:

$mkdir *zal zal/intel zal/sm4 zal/multibus*| zal/sm4/camac koss kos/intel kos/kswdb| kos/intel/8080 kos/intel/8086 schw| schw/8080 schw/cross

Oczywiście, przy próbie utworzenia skorowidza *kos/intel* w powyższym ciągu nastąpi wyprowadzenie meldunku o błędzie:

mkdir: cannot access kos

ponieważ uprzednio nie utworzono skorowidza *kos* lecz (pomyłkowo) koss.

rmdir

W związku z poprzednim meldunkiem o błędzie należy usunąć skorowidz *koss* (nie jest to bezwzględnie konieczne, lecz pożądane, gdyż nikt go nie będzie dalej używał), używając polecenia:

$rmdir *koss*

$

Usunięcie skorowidza przy użyciu polecenia **rmdir** (ang. remove a directory) jest możliwe tylko wtedy, gdy skorowidz jest pusty, tzn. nie zawiera żadnego pliku, a ponadto nie jest skorowidzem roboczym (ang. working directory), z którego wydano polecenie. Przy próbie usunięcia niepustego skorowidza, np. utworzonego wcześniej skorowidza *zal*, otrzymamy meldunek o błędzie, tzn.:

$rmdir *zal*
rmdir: *zal* not empty

$

ponieważ zawiera on trzy skorowidze podrzędne – *sm4, intel* i *multibus*. Proba usunięcia skorowidza, którego jeszcze nie utworzono, np. *kos,* spowoduje również wyprowadzenie meldunku o błędzie, tzn.:

$rmdir *kos*
rmdir: *kos* non-existent

$

pwd

Zakładając, że utworzono żądane skorowidze, tj. *zal, kos* i *schw* oraz wszystkie podległe, można zażądać określenia bieżącego skorowidza, czyli tzw. skorowidza roboczego. Do tego celu służy polecenie **pwd** (ang print working directory). W naszym wypadku będzie to skorowidz nadrzędny (ang. parent) względem pierwotnie utworzonych, tj.:

$pwd
/usr

$

Po wydaniu polecenia **pwd** następuje wyprowadzenie pełnej nazwy ścieżki (ang. path name) prowadzącej do skorowidza roboczego począwszy od skorowidza pierwotnego (*root*). Do skorowidza *zal* od skorowidza pierwotnego, oznaczonego zawsze niejawnie znakiem "/", prowadzi ścieżka przez skorowidz *usr*.

cd

Rzeczą równie istotną jak znajomość bieżącego skorowidza jest możliwość jego zmiany. Do tego celu służy polecenie **cd** (ang. change directory). Przykładowo, chcąc zmienić skorowidz roboczy z **usr** na 8086, należy wydać polecenie:

$cd /usr/kos/intel/8086

Oczywiście, wydanie tego polecenia spowoduje wyprowadzenie meldunku:

/usr/kos/intel/8086: bad directory

ponieważ wcześniej nie utworzono poprawnie ani skorowidza *kos*, ani żadnego ze skorowidzów podległych. Na ogół nie wiadomo z góry, jaka jest przyczyna meldunku:

bad directory

ponieważ równie dobrze może to być spowodowane próbą odwołania się do zwykłego pliku zamiast do skorowidza, jak również próbą odwołania się do skorowidza bez prawa dostępu do niego.

Pierwszą rzeczą, którą warto zrobić po skutecznym lub nieskutecznym wykonaniu polecenia cd, jest wydanie polecenia **pwd** w celu sprawdzenia, w jakim skorowidzu aktualnie pracujemy.

Przykładowo, wydając polecenie:

$cd */usr/zal/sm4/camac*

$

a następnie polecenie:

$cd *../../../schw/cross*

powodujemy przejście do skorowidza *cross,* co można sprawdzić poleceniem

$pwd
/usr/schw/cross

$

Wydając polecenie cd bez żadnego argumentu, powodujemy przejście do skorowidza macierzystego, tzn. przypisanego użytkownikowi po zarejestrowaniu się w systemie, np.:

$cd
$pwd
/usr/zal

PODSTAWY OPEROWANIA PLIKAMI

Poleceń tego rodzaju jest najwięcej, gdyż praktycznie każde polecenie systemu UNIX jest związane z określonym plikiem.

ls

Polecenie **ls** (ang. list sorted) jest najczęściej używanym poleceniem systemu UNIX. Powoduje wyprowadzenie nazw plików za-

[1] Kreska pionowa (ang. backslash) oznacza przedłużenie wiersza

wartych w skorowidzu roboczym lub innym, wymienionym z nazwy. Przykładowo, wydanie tego polecenia ze skorowidza *zal,* spowoduje wyprowadzenie następującej informacji:

$ls
intel
multibus
sm4
$

W poleceniu tym często używa się opcji zaznaczonych myślnikiem „–". Opcje zostaną omówione dokładniej przy opisie powłoki (ang. shell) lecz już teraz należy zwrócić uwagę na dwie z nich:

– a (ang. all, wszystkie pliki)
– l (ang. long, pełna informacja).

Posługując się omawianym poleceniem z wymienionymi opcjami, można określić pełną zawartość skorowidzów *usr, bin* i *etc* np. w pierwszym wypadku:

$cd /
$pwd
/usr
$ls -al

może spowodować wyprowadzenie następującej informacji:

```
.
..
drwxr-xr-x  2 schw   10  Sep01  6:37 schw
drwxr-xr-x  2 root  350  Jul22 12:00 games
drwxr-xr-x  2 root  450  Jul22 12:01 include
drwxr-xr-x  2 kos    15  Aug16 18:08 kos
drwxr-xr-x 20 root  300  Jul22 12:02 man
drwxr-xr-x 20 root  320  Jul22 12:05 src
drwxr-xr-x  2 root  120  Jul22 12:06 sys
drwxr-xr-x  2 zal    15  Oct01 11:20 zal
```

Oprócz trzech skorowidzów założonych przez użytkowników *(schw, kos* i *zal),* w skorowidzu *usr* znajdują się inne, których nazwy wymieniono w ostatniej kolumnie:

- *games,* zawierający programy gier komputerowych;
- *man,* zawierający podręcznik systemu UNIX (Unix Programmer's Manual);
- *src,* zawierający teksty źródłowe programów usługowych;
- *sys,* zawierający teksty źródłowe samego systemu UNIX.

Pierwsza kolumna powyższego wydruku zawiera informację dotyczącą praw dostępu do plików. Wynika z niej, że wszystkie te pliki są skorowidzami (litera *d* na pierwszej pozycji), a prawo do ich odczytania i dostępu do ich zawartości mają wszyscy użytkownicy, natomiast prawo zapisu przysługuje tylko właścicielom, których nazwy podano w trzeciej kolumnie wydruku. W drugiej kolumnie znajduje się liczba łączników do wymienionych skorowidzów. Wynosi ona 2, co znaczy, że dany skorowidz jest dołączony do skorowidza nadrzędnego (..) i siebie samego (.). W czwartej kolumnie zawarta jest informacja dotycząca wielkości pliku w bajtach, a w przedostatniej datownik, tj. czas utworzenia pliku.

Podobnie można sprawdzić zawartość skorowidza *bin,* który zawiera w praktyce ponad 150 programów usługowych. Poniżej wymieniono jedynie programy omówione w tej części artykułu.

$cd *../bin*
$pwd
/bin
$ls -l

```
-rwx--x--x 1 bin  9200 Mar08 19:35 cat
-rwx--x--x 1 bin  8200 Mar08 19:35 cd
-rwx--x--x 1 bin  8150 Mar08 19:35 chmod
-rwx--x--x 1 bin  9500 Mar08 19:35 cp
-rwx--x--x 1 bin 12500 Mar08 19:36 file
-rwx--x--x 1 bin 14300 Mar08 19:36 find
-rwx--x--x 1 bin  6600 Mar08 19:37 ln
-rwx--x--x 1 bin 21100 Mar08 19:37 ls
-rwx--x--x 1 bin  8600 Mar08 19:38 mkdir
-rwx--x--x 1 bin 10200 Mar08 19:38 mv
-rwx--x--x 1 bin  8400 Mar08 19:38 pwd
-rwx--x--x 1 bin 10150 Mar08 18:38 rm
-rwx--x--x 1 bin  8700 Mar08 18:38 rmdir
-rwx--x--x 1 bin  9200 Mar08 18:39 tail
-rwx--x--x 1 bin  8500 Mar08 18:39 wc
```

Wszystkie te pliki są zwykłe, tzn. są programami wykonywalnymi (myślnik „–" na pierwszej pozycji pierwszej kolumny wydruku) przez wszystkich użytkowników (atrybut x dla wszystkich rodzajów użytkowników), a liczba łączników we wszystkich przypadkach wynosi 1, tzn. że pliki nie są dołączone do żadnego innego skorowidza. Nazwy poszczególnych plików stanowią odpowiednie polecenie systemowe i zostaną omówione poniżej.

Oprócz skorowidzów systemowych *usr* i *bin* warto znać zawartość skorowidza *etc,* będącego na ogół pod kontrolą zarządcy systemu (ang. system manager):

$cd *../etc*
$pwd
/etc
$ls -l

```
-rw-r--r-- 1 bin      30 Jul22 16:50 ident
-rw-r--r-- 1 bin    8130 Jul22 16:50 mkfs
-rw-r--r-- 1 bin      30 Jul22 16:55 motd
-rw-r--r-- 1 bin    9259 Jul22 16:55 mount
-rw-r--r-- 1 root   2460 Jul22 18:20 passwd
-rw-r--r-- 1 bin   42020 Jul22 16:20 termcap
-rw-r--r-- 1 bin    9500 Jul22 16:55 umount
```

$

Rola wymienionych w skorowidzu etc programów mount i umount zostanie omówiona w dalszej części artykułu.

cat

Wykonanie polecenia **cat** (ang. concatenate)[2] polega na sklejeniu plików będących argumentami i wyprowadzeniu ich całej zawartosci na standardowe urządzenie wyjściowe, zazwyczaj na ekran monitora. Przykładowo, można pokazać swojemu szefowi, że dziś nie zdążymy wykonać pracy, którą nam zlecił, wyprowadzając zawartość pliku **motd** (jesteśmy własnie w skorowidzu zawierającym ten plik):

$cat motd

Dziś (12 sierpnia 1985), jak w co drugą środę, o godz. 12.00 nastąpi zawieszenie pracy komputera na 8 godzin ze względu na konieczne zabiegi konserwacyjne. Przepraszamy.

$

Jeżeli użyjemy polecenia **cat** do pliku, który nie istnieje lub dla którego nie mamy prawa odczytu, to otrzymamy w odpowiedzi meldunek o błędzie, np.

$cat *zal/informatyka*
cat: cannot open *informatyka*
$

Dlatego też przed wykonaniem tego polecenia warto sprawdzić, jaki jest bieżący skorowidz (poleceniem **pwd**) i jaka jest jego zawartość (poleceniem **ls**).

Należy tez pamiętać, że podstawowym celem omawianego polecenia jest sklejanie plików, a wyprowadzenie ich zawartości stanowi, do pewnego stopnia, efekt uboczny. Dlatego nie należy oczekiwać sensownych wyników przy wyświetlaniu zawartości plików nietekstowych.

Jeżeli polecenie cat nie ma żadnych argumentów, to jego działanie polega na wprowadzeniu danych ze standardowego pliku wejściowego (np. terminala) i wyprowadzeniu ich na standardowy plik wyjściowy (którym może być ten sam terminal), np.:

$cat
To jest moja prywatna papuga.
^D
To jest moja prywatna papuga.
$

Wprowadzenie tekstu zakończono znakiem control-D (^D) oznaczającym EOT (End Of Text).

Specyfikując plik wyjściowy, polecenia **cat** można używać do tworzenia plików, np.:

$cd *../usr/zal/sm4/camac*
$cat *pierwsza_liga*
Górnik-Legia 7:0
Wisła-Radomiak 1:5
^D
$

Oczywiście, bardziej skuteczne jest tworzenie plików za pomocą edytora. Czytelnikom zalecamy wykonanie kopii dowolnego pliku przy użyciu polecenia **cat.**

cdn.

JANUSZ ZALEWSKI

[2] Nawet po dłuższej pracy z systemem UNIX trudno zapamiętać funkcje niektórych poleceń na podstawie ich dziwnych nazw. W przypadku polecenia cat moj prywatny przepis na zapamietanie jego funkcji polega na przyjęciu, że źródłosłowem angielskim nazwy jest zwrot connect and type – w skrócie cat.

UNIX w drodze po sukces

Era komputerów osobistych budowanych na bazie mikroprocesorów 8-bitowych spowodowała rozwój jednoprogramowych systemów operacyjnych, wśród których niewątpliwie przodował system CP/M.
Zastępowanie układów 8-bitowych układami 16- i 32-bitowymi powoduje nie tylko ilościowy wzrost możliwości komputerów osobistych. Przestają one być sensu stricto osobiste, a stają się systemami wielodostępnymi. Aby było to możliwe, wręcz rewolucyjnej zmianie muszą ulec systemy operacyjne.
Często mniej doświadczony użytkownik nie zauważa dużych różnic między obsługą terminala systemu wielodostępnego i posługiwaniem się 8-bitowym komputerem osobistym.
Z dość dużym prawdopodobieństwem można obecnie prognozować, że miejsce systemu CP/M dla komputerów 16- i 32-bitowych zajmie system UNIX i różne jego mutacje. Potwierdzeniem tej tezy niech będzie poniższa tabela, w której podajemy charakterystykę kilkunastu mikrokomputerów oferowanych przez czołowych producentów.

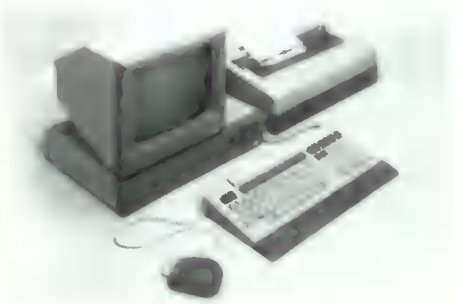
micro

Producent'	Komputer	Procesor		System operacyjny
		hazwa	liczba bitów	
AT&T	rodzina systemów 3B	WE 32100 WE 32106	32	UNIX System V
Bull	SPS 9/40 SPS 9/69		32 32	ROS (wymienny z UNIX System V)
Bull	Questar 400	Intel 80186	16	MS-DOS lub Prologue
Commodore	CBM 900 II	Zilog Z8000	16	Coherent (wymienny z UNIX System V)
Compaq	Deskpro 286 Portable 286 (przenośny)	Intel 80286	16	MS-DOS 3 Xenix
Cromemco	CS 100 CS 300	MC 68000	16/32	UNIX System V lub Cromix
NCR	miniTower Tower XP NCR 300	MC 68010 Intel 80186	16/32 16	UNIX System V CTOS (wymienny z CP/M 86 i MS-DOS)
Rair	MiniMicro	Intel 80286	16	Concurrent-DOS
Seikosha	Seiko System 286	Intel 80286	16	Oasis lub Thoroughbred-OS lub Xenix
Siemens	PC MX PC X Sinix PC (przenośny)	Intel 80186	16	Sinix
Sperry	Seria 500	MC 68000 MC 68010	16/32 16/32	UNIX System V
Triumph Adler	System M 32	MC 68000	16/32	Tanix
Texas Instruments	Business-Pro	Intel 80286	16	MS-DOS 3 lub Xenix 3
Wang	VS 15	Intel 80286	16	

1) Układ zarządzania pamięcią wirtualną
2) Jednocześnie można podłączyć maksymalnie 12 stanowisk

Standard Centronics

Komputer osobisty PC-MX firmy Siemens



System M33 firmy Triumph-Adler

Pamięc operacyjna (pojemność w MB)	Pamięć masowa (pojemnośc w MB)	Liczba stanowisk
–	–	–
	⩽ 8000[1]	
–	–	8
0,5...2	–	7
⩽ 8,2 ⩽ 2,6	70 20	10...16
0,5...2	50...1300	
0,5...2 1...8 1...8	– 26...170 –	8...16[2] –
1	25	–
1...4	20...50	12
0,5	–	4
1...16	–	8...64
0,5...2	71	–
3...15	72	8
4	–	30

Uwaga: kreska w tabeli oznacza brak danych

Drukarka to bardzo istotny element systemu komputerowego. Potwierdzą to wszyscy, którzy widzą w posiadanym mikrokomputerze coś więcej niż inteligentną grę telewizyjną. Chociaż jest to wyjątkowo drogi sprzęt (często droższy od samego komputera) – trudno sobie bez niego poradzić.

Komputera, do którego nie można przyłączyc drukarki, nikt nie będzie traktował poważnie. Dlatego też nawet producenci komputerów-zabawek proponują przynajmniej substytut tego urządzenia. Proste mikrokomputery ze względów ekonomicznych nie mają specjalnego gniazda, umożliwiającego przyłączenie drukarki. Trzeba więc wykorzystać gniazdo uniwersalne (najczęściej jest to wyprowadzenie wszystkich szyn mikroprocesora), co powoduje, że drukarka musi być ściśle dopasowana do wewnętrznej architektury danego mikrokomputera. Oczywiście nie pasuje ona wtedy do żadnego innego komputera i odwrotnie: drukarki firm nie związanych z producentem komputera, zazwyczaj nie dają się do niego przyłączyć.

Producenci o większych aspiracjach wyposażają komputer w osobne gniazdo do przyłączenia drukarki. Gniazdo to i niezbędne do jego obsługi układy elektroniczne to tzw. *interfejs*. W tym przypadku użytkownik ma już znacznie więcej swobody, gdyż interfejs zazwyczaj spełnia wymagania standardu określonego mianem **Centronics.** Również drukarki, nie związane z określonym komputerem, zwykle są wyposażone w interfejs Centronics. Dzięki temu można wybierać między drukarkami tanimi, o ograniczonych możliwościach i droższymi, które ładniej piszą lub precyzyjniej rysują, czasem nawet w kolorze. Dla uzyskania swobody wyboru opłaca się więc nawet dobudować do prostego mikrokomputera przystawkę interfejsu Centronics.

Standard interfejsu Centronics pozwala uzgodnić sposób elektrycznego połączenia urządzeń i zsynchronizować przesyłanie informacji. Poszczególne sygnały interfejsu zestawiono w tabeli. Interfejs Centronics pozwala na równoczesne przesyłanie danych po ośmiu liniach. Dlatego też jest określony mianem *interfejsu równoległego* (w odróżnieniu od interfejsów szeregowych, w których informacja jest przesyłana po jednej linii).

Informacje są przesyłane w znormalizowanym kodzie ASCII[1], w którym każdej literze, cyfrze lub symbolowi jest przypisana określona liczba siedmiobitowa. Spowodowało to, że niektórzy producenci komputerów zredukowali zestaw linii danych z ośmiu do siedmiu. Dopóki drukarki nie dysponowały możliwościami graficznymi, nie powodowało to żadnych utrudnień. Obecnie w wielu drukarkach dodatkowy bit został wykorzystany do wprowadzenia zestawu znaków graficznych lub symboli matematycznych, których nie przewidziano w kodzie ASCII. Jeżeli komputer steruje tylko siedmioma liniami danych, to dodatkowego zestawu nie można wykorzystać. W ulepszonych modelach drukarek rozwiązano jednak tę niedogodność: przesyłając specjalny kod sterujący można spowodować, by następne informacje były traktowane tak, jakby na ósmą linię podawano logiczne jeden (lub zero), mimo że linia ta nie jest sterowana.

W drukarkach zwykle jest wykorzystywane 36-stykowe gniazdo firmy Amphenol o numerze 57-30360. We wspomnianej tabeli podano również znaczenie sygnałów przypisanych poszczególnym wyprowadzeniom. Od strony komputera sprawa niestety nie jest rozwiązana równie elegancko. Wykorzystywane są różne typy łączówek (najczęściej tzw. łącza krawędziowe), zestaw sygnałów jest często zredukowany, a ponadto nie istnieje jednolity sposób przypisania sygnałów do wyprowadzeń. Zazwyczaj jednak producent oferuje przewód, który doprowadza właściwe linie do wtyku firmy Amhenol. Niestety przewód tego typu kosztuje kilkanaście dolarów i jeżeli dysponujemy odpowiednimi łączówkami, znacznie taniej będzie przygotować go samemu. W załączonej do komputera – każdej poważnej firmy – dokumentacji można znaleźć opis łączówki interfejsu drukarki i wystarczy porównać go z naszą tabelą, by bez trudu określić wszystkie połączenia. Nie należy przy tym rozpaczać, jeżeli nie znajdziemy któregoś z sygnałów wymienionych w tabeli, nie wszystkie bowiem linie są bezwzględnie konieczne do zrealizowania zadowalającej współpracy komputera z drukarką.

Jeżeli do współpracy z drukarką wykorzystujemy interfejs, wykonany w postaci przystawki do mikrokomputera, standard Centronics pozwala na zasilanie układów przez drukarkę. Służy do tego celu wyprowadzenie nr 18. Jest to bardzo cenna możliwość, gdyż zasilacze komputerów bardzo często pracują na granicy swoich możliwości.

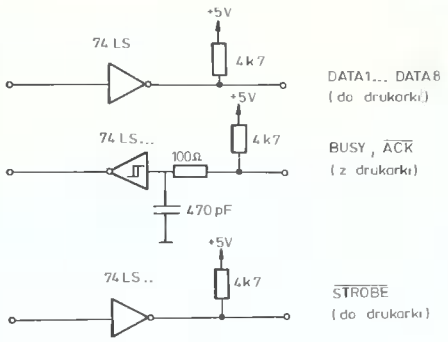
Warto jeszcze zwrócić uwagę na sposob łączenia masy. Celem zmniejszenia podatności na zakłócenia proponuje się, aby każdy przewód sygnałowy został skręcony z przewodem masy sygnałowej. Każdy sygnał ma więc „własną" masę, która oczywiście jest połączona elektrycznie z pozostałymi.

[1] ASCII = American Standard Code for Information Interchange – amerykański znormalizowany kod do wymiany informacji

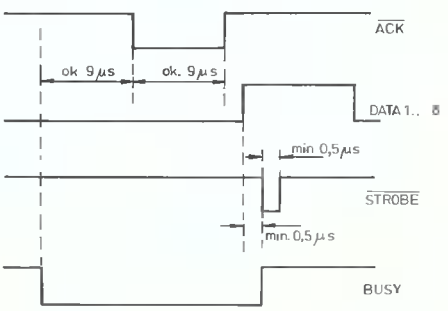
Ponadto, każda linia sygnałowa powinna być „podciągnięta" rezystorem (4,7 KΩ) do napięcia +5V. Na wejściach układów buforujących linie kontrolno-sterujące należy umieścić dolnoprzepustowy filtr RC (rys. 1).

Przesyłając dane do drukarki należy na linii STROBE/ wymusić poziom niski. Impuls nie powinien być krótszy niż 0,5 µs. Drukarka odpowiada wymuszając poziom wysoki na linii BUSY (rys. 2). Przesłanie następnego bajtu danych nie powinno jednak nastąpić wcześniej niż po 18 µs od powrotu linii BUSY do poziomu niskiego. Jeżeli obsługa interfejsu drukarki jest realizowana przez przerwania, może okazać się celowe wykorzystanie impulsu pojawiającego się na linii ACK/. Dotyczy to systemów, w których przerwanie jest generowane nie przez poziom, ale przez zbocze sygnału.

ANDRZEJ J. PIOTROWSKI



Rys. 1. Przykład buforowania linii interfejsu Centronics



Rys. 2. Przebiegi czasowe przy przesyłaniu danych

Sygnały interfejsu Centronics (w nawiasach podano numer wyprowadzenia masy sygnałowej)

Nr wyprowadzenia	Mnemonik linii	Opis funkcjonalny linii interfejsu
1 (19)	STROBE/	impuls powodujący wpisanie danych do bufora drukarki
2 do 9 (20 do 27)	DATA1 ... DATA 8	osiem linii danych; stan logiczny 1 jest sygnalizowany wymuszeniem poziomu wysokiego
10 (28)	ACK/	ujemny impuls o długości ok. 5 µs, generowany po wprowadzeniu danych przez drukarkę
11 (29)	BUSY	poziom wysoki sygnalizuje, że drukarka nie zakończyła jeszcze wprowadzania
12 (30)	PAPER END	poziom wysoki sygnalizuje brak papieru
13	SELECTED	poziom wysoki sygnalizuje gotowość drukarki do współpracy z komputerem
14 i 15	-	nie wykorzystywany
16	SIGNAL GND	masa sygnałowa
17	CHASSIS GND	uziemienie
18	+5V DC	napięcie zasilające +5V (ok. 50 mA)
31	INPUT PRIMPE/	poziom niski zeruje układy drukarki
32	ERROR/	sygnalizacja nieprawidłowości w przesyłanej informacji
33	EXT GND	masa zewnętrzna
34 i 35	-	nie wykorzystywany
36	-	poziom niski

Mimo że niemal wszystko w komputerze odbywa się za pośrednictwem pozostającego w tle systemu operacyjnego, użytkownik poświęca mu niewiele uwagi. Tymczasem wybór systemu operacyjnego decyduje o późniejszej przydatności komputera w praktyce. Warto więc w porę zapoznać się z tym zagadnieniem. W ten sposób uniknie się późniejszych rozczarowań, gdy komputer nie potrafi tego czy innego, ponieważ system operacyjny nie spełnia odpowiednich warunków. Warto wiedzieć co nieco o systemach operacyjnych, aby nie stracić gruntu pod nogami przy ewentualnej późniejszej zmianie systemu, także wówczas, gdy nabyło się gotowe i proste w użyciu rozwiązanie (komputer + programy użytkowe).

System operacyjny jest częścią tak zwanego oprogramowania systemowego (ang. system software). Pod tym pojęciem rozumie się wszystkie programy służące do wprowadzania i uruchamiania programów użytkowych. Poszczególni dostawcy w różny sposób dokonują podziału oprogramowania systemowego, najczęściej jednak podział ten jest następujący:

- system operacyjny
- programy usługowe (ang. utilities)
- programy tłumaczące (translatory)
- programy narzędziowe (ang. software tools)
- programy komunikacyjne.

W zestawie nie znalazły się języki programowania, ponieważ nie są to programy, lecz zbiory przepisów, do których musi stosować się programista przy pisaniu programu, aby był on zrozumiały dla programu tłumaczącego (translatora). Właściwy dla każdego języka program tłumaczący przekształca napisany przez użytkownika program użytkowy w sekwencję rozkazów zrozumiałych dla komputera.

CZYM JEST SYSTEM OPERACYJNY?

Przytoczmy żartobliwe pytanie z pewnej rozprawy naukowej o systemach operacyjnych: *Czym jest właściwie słoń?*. Prawidłowa odpowiedź brzmi: *Słoń jest to mysz z systemem operacyjnym.* Stwierdzenie to stanie się zrozumiałe, gdy uświadomimy sobie, że systemy operacyjne dla dużych komputerów mogą się składać z wielu milionów rozkazów.

System operacyjny jest do pewnego stopnia „pośrednikiem" między jednostką centralną (właściwym komputerem), pamięcią operacyjną a dołączonymi urządzeniami zewnętrznymi oraz między sprzętem (ang. hardware) a programami użytkowymi. System operacyjny składa się z dużej liczby programów sterujących, służących – na przykład – do przesyłania danych między różnymi urządzeniami zewnętrznymi (sterowanie wejścia-wyjścia), do przydzielania miejsca w pamięci operacyjnej i w pamię-

Ukryte gwiazdy

ciach zewnętrznych (dyskietkach lub dyskach sztywnych), a także do obsługi przerwań (ang. interrupts).

Systemy operacyjne mają z reguły strukturę modularną. Oznacza to, że składają się z wielu samodzielnych części, zwanych modułami. Wynika to z faktu, że łatwiej jest napisać i przetestować poszczególne moduły oraz nie trzeba wprowadzać do pamięci operacyjnej od razu całego systemu operacyjnego, który często w całości by się w niej nie zmieścił. Dlatego system operacyjny znajduje się zwykle na dysku lub dyskietce, zaś do pamięci operacyjnej wprowadzane są tylko moduły potrzebne w danej chwili.

Pewna część systemu operacyjnego pozostaje jednak stale w pamięci operacyjnej, wykonując podstawowe działania sterujące i ładując potrzebne moduły. Jest ona często na tyle duża, że przy ograniczonej pojemności pamięci operacyjnej uniemożliwia wykonywanie dużych programów użytkowych.

Proste systemy operacyjne zawierają procedury sterujące, które służą do wymiany danych z urządzeniami wejściowymi i wyjściowymi komputera, a także procedury realizujące różne metody dostępu do zbiorów danych, zapisanych w pamięciach zewnętrznych.

Rozbudowane systemy zapewniają więcej: przejmują całe zarządzanie danymi (ang. data management), dzięki czemu – na przykład – przy instalowaniu programu nie trzeba od razu ustalać, na jakim nosniku będą zapamiętane dane. Obecnie zarządzanie danymi odbywa się najczęściej przy użyciu specjalnych baz danych i systemów zarządzania bazami danych.

Jak już wspomnieliśmy, do systemu operacyjnego należy także obsługa przerwań. Sygnały przerwań pojawiają się nie tylko po wykryciu błędów lub sytuacji awaryjnych (np. zacięcie się papieru w drukarce), ale także w czasie normalnej pracy. Na przykład, gdy jednostka sterująca dysków magnetycznych przekazuje meldunek, że wykonała zadanie i jest „wolna", lub gdy terminale ekranowe sygnalizują, że są gotowe do przesyłania danych do pamięci operacyjnej. Użytkownik niczego nie zauważa, gdyż takie przerwania obsługuje specjalna procedura (ang. interrupt handler) w kolejności zależnej od priorytetów zgłoszeń.

Obsługa przerwań jest szczególnie ważna przy pracy wieloprogramowej (ang. multiprogramming) oraz pracy z podziałem czasu (ang. time sharing). Przy pracy wieloprogramowej komputer wykonuje wiele programów „jednocześnie". Przerywa on wykonywanie jednego programu w czasie operacji wejścia-wyjścia, aby w tym czasie wykonać rozkazy innego programu. Dzięki temu czas pracy komputera jest lepiej wykorzystany. Przykładowo, w czasie gdy program *A* wysłał wiersz na drukarkę i czeka na zakończenie drukowania, aby móc wysłać wiersz następny, program *B* może zapisać rekord danych w pamięci.

W systemie z podziałem czasu komputer przydziela programom *A, B, C* z góry określone przedziały czasu. W ten sposób przez kilka milisekund pracuje program *A*, przez kilka następnych program *B*, a potem program *C* i znów program *A* itd. Żaden z użytkowników siedzących przy terminalach nie zauważa, że ma do dyspozycji tylko określone odcinki czasu, lecz wydaje mu się, że komputer pracuje tylko dla niego. Oczywiście użytkowników nie może być zbyt wielu, zaś komputer musi być dostatecznie szybki.

Prawdziwy system wielodostępny powinien zarządzać przydziałem pamięci operacyjnej i dokonywać w tej pamięci wymiany (ang. swapping) zadań programowych (ang. tasks). Obecnie w mikrokomputerach stosuje się również nieco mniej wydajne systemy, pozwalające jednak na „jednoczesną" pracę kilku programów. Poszczególnym programom przydzielane są stałe obszary pamięci operacyjnej zwane partycjami (ang. partitions). Organizowanie drukowania (ang. spooling od spool – szpulka) np. polega na przejściowym umieszczaniu w pamięci zewnętrznej wydruków, skąd są potem odczytywane i przenoszone na papier.

Płynne wznowienie realizacji programu po jego awarii jest ważne w każdym systemie operacyjnym, ale w systemie wielodostępnym ma znaczenie podstawowe. Płynność oznacza w tym wypadku możliwość dokładnego ustalenia, które dane zostały przetworzone i zapisane w pamięci bezbłędnie. Ponownego wprowadzenia wymagają tylko dane, na które mogła mieć wpływ awaria. Wznowieniem programu w systemach operacyjnych zajmują się często tak zwane procedury restartu (ang. restart routines).

Równie ważne jest istnienie w systemie operacyjnym mechanizmów zabezpieczających, które zapewniają dostęp do określonych danych tylko określonym użytkownikom. Rozróżnia się ochronę przed zapisem i przed odczytem. Nieuprawniony użytkownik może wprawdzie czytać dane chronione przed zapisem, lecz nie może ich zmieniać, a danych chronionych przed odczytem nie może nawet odczytać. Zabezpieczenia takie mogą istnieć na różnych poziomach: chronione mogą być całe zbiory, poszczególne rekordy lub tylko niektóre pola w rekordach. Dostęp do danych z reguły ma tylko ten użytkownik, który działa z terminala przyporządkowanego do określonego programu i zna odpowiednie hasło.

Wielu pracowników biurowych już dzisiaj ma na swym biurku terminal (monitor ekranowy) przyłączony do komputera centralnego. Przy takim rozwiązaniu mogą oni wykonywać prace realizowane przez duże komputery. Często istnieje jednak potrzeba automatyzacji prac bezpośrednio związanych ze stanowiskiem pracy. Instaluje się wtedy na biurkach mikrokomputery, które mogą działać niezależnie lub komunikując się z komputerem centralnym jak terminal.

Zastępowanie terminali przez mikrokomputery, które są coraz tańsze, zmusza oczywiście producentów terminali do podjęcia odpowiednich środków zaradczych. Umieszczają oni w swych wyrobach mikroprocesory i zaopatrują je w „systemy opracyjne komputerów osobistych", czyniąc terminale urządzeniami coraz bardziej inteligentnymi.

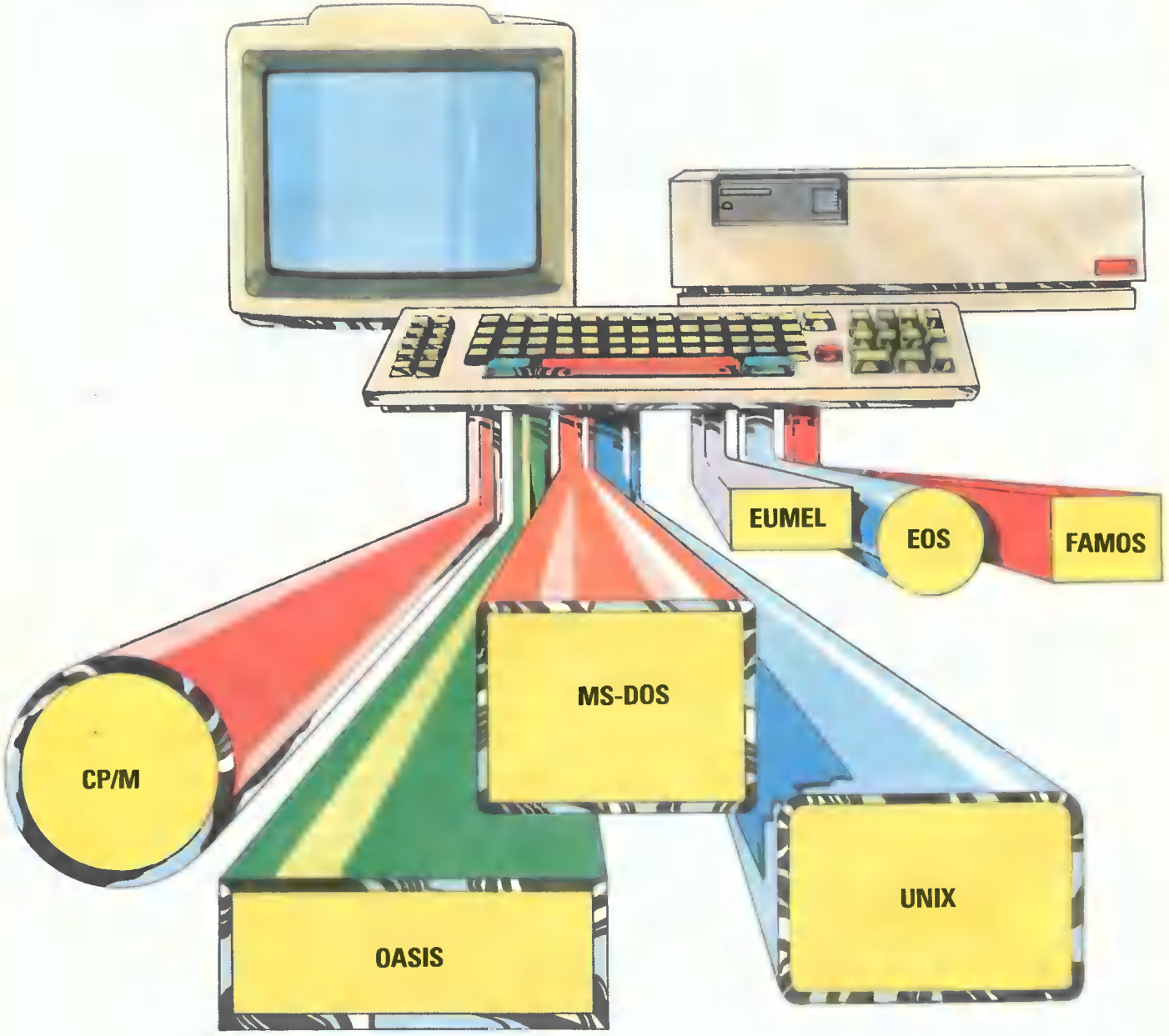
W obu wypadkach pozostaje jednak do rozwiązania problem polegający na tym, że gdy użytkownik chce w czasie wykonywania własnego programu wykorzystywać dane z komputera centralnego, oba systemy operacyjne (mikrokomputera i komputera centralnego) muszą się z sobą komunikować. Żaden z nich jednak nie jest do tego przygotowany. Konieczne są więc dodatkowe programy emulacyjne i programy transmisji danych.

Sposrod systemów operacyjnych wielodostępnych zdecydowanie na pierwsze miejsce wysunął się obecnie system Unix. Większość producentów oferuje go w swych zestawach przynajmniej jako dodatkową opcję. Najbardziej rozpowszechnioną dla mikrokomputerów odmianą systemu Unix jest bez wątpienia Xenix, wywodzący się z wersji Unix System III i opracowany przez firmę Microsoft.

Od chwili, gdy w 1969 r. firma AT&T opracowała w Bell Laboratories system Unix,powstało wiele jego wersji. Początkowo AT&T dostarczała „gotowy" system operacyjny tylko dla komputerów firmy DEC, a inni producenci komputerow musieli sami lub przy pomocy producentów oprogramowania dopasowywać go do swych wyrobów. Wkrótce AT&T zaoferowała wersje dostosowane do procesorów firm Intel, Motorola, National Semiconductor i Zilog.

Dużą popularność wśród producentów komputerów zawdzięcza Unix różnorodności wersji, co ułatwia dopasowanie go do dowolnego mikrokomputera. Z drugiej strony, i jest to jego słaby punkt, istnieje dotąd niewiele programów użytkowych dostosowanych do tego systemu. Tu właśnie tkwi przyczyna różnych jego ocen. Rynek oprogramowania użytkowego współpracującego z systemem Unix, stopniowo wzrasta gdyż:

- firma AT&T lansuje go z dużym rozmachem
- powstają nowe wersje systemu Unix ułatwiające programowanie, dzięki czemu coraz więcej producentow oprogramowania jest zainteresowanych odpowiednimi programami użytkowymi



■ producenci pracują nad tym, by system służył przede wszystkim użytkownikowi, usuwają więc procedury dla niego nie istotne oraz poprzez zastosowanie programu nadrzędnego czynią sam system operacyjny „niewidocznym".

Z powodu różnorodności wariantów użytkownik nie zawsze może się łatwo przestawić z jednego komputera, pracującego pod nadzorem systemu Unix, na inny. Problem ten dotyczy jednak także innych systemów operacyjnych. Niewątpliwą zaletą systemu Unix jest możliwość jego zainstalowania także na dużych komputerach. Do pewnego stopnia gwarantuje to, że Unix będzie dalej rozszerzony poprzez wersję 32-bitową do 64 -bitowej. Jego wadą, z punktu widzenia użytkownika, jest stosunkowo duże zapotrzebowanie pamięci. Niedogodność ta traci na znaczeniu, w miarę spadku cen pamięci.

Innym systemem operacyjnym, mającym zastosowanie zarówno w konfiguracji jedno-, jak i wielodostępnej jest system OASIS, amerykańskiej firmy Phase One Systems Inc., przeznaczony dla mikrokomputerów 8- i 16-bitowych. Oprócz ogólnie spotykanych narzędzi programowych, takich jak na przykład generator programów, użytkownik ma tu do dyspozycji relacyjną bazę danych z językiem zapytań (ang. query language).

Wielu producentów zapewnia, że dla użytkowników systemu operacyjnego CP/M otwierają się wspaniałe perspektywy, gdyż programy użytkowe pracujące pod nadzorem tego systemu można kupić niemal na każdym rogu ulicy. Perspektywy te nie są jednak pozbawione również cieni. Wprawdzie CP/M zainstalowano na świecie już na przeszło 500 tys. mikrokomputerów – ale w różnych wersjach!

Początkowo CP/M był jedynie programem zarządzającym zbiorami, w mikroprocesorowym systemie uruchomieniowym opracowanym przez pracownika firmy Intel. Pojawił się on już w 1975 r. Z upływem lat został rozszerzony i ulepszony. Obok systemu operacyjnego CP/M, dostosowanego tylko do mikroprocesorów 8-bitowych, pojawił się opracowany przez firmę Digital Research system operacyjny CP/M 86, przeznaczony do pracy z mikroprocesorami 16-bitowymi (Intel 8086) oraz 8/16-bitowymi (Intel 8088). Oba systemy są jednak systemami dla jednego użytkownika (ang. single user systems) i pracują tylko na instalacjach z jednym terminalem.

W tym czasie powstał jednak Concurrent CP/M-86 (CCP/M-86), czyli wersja wielozadaniowa. Także Turbo-DOS, początkowo opracowany specjalnie dla mikroprocesora Z80-A, może być zainstalowany w syste-

mach z jednym lub wieloma terminalami i współpracowac z siecią lokalną.

Na długi czas CP/M stał się swoistym standardem przemysłowym wśród systemów operacyjnych. Wraz z wejściem na rynek mikrokomputerowy firmy IBM ustalił się drugi „standard": system operacyjny MS-DOS, który początkowo niedoceniany, cieszy się obecnie coraz większą popularnością.

MS-DOS jest wprawdzie systemem jednostanowiskowym, ale dla procesorów 16-bitowych. Został opracowany przez Tima Patersona dla procesora Intel 8086. Na rynku pojawił się najpierw pod nazwą SCP DOS-86. Później prawa do tego systemu nabyła firma Microsoft, nazywając go MS-DOS. Kiedy w końcu firma IBM zdecydowała się zainstalować go na swoim komputerze osobistym IBM PC, system otrzymał nazwę PC-DOS.

RÓŻNORODNOŚĆ ZAMIAST STANDARDU

Czytelnik nie powinien dać się zmylić użytemu tu słowu *standard*, ponieważ nie istnieje żadna oficjalna norma dla systemów operacyjnych. Słowo *standard* nie jest dokładne choćby z tej przyczyny, że prawie każdy system operacyjny ma wiele odmian i wersji. Którą wersję uznać więc za standardową? Różnorodność systemów operacyjnych wcale nie jest korzystna dla potencjalnego użytkownika. Na ogół potrzebuje on określonej instalacji sprzętowej z ustalonymi programami użytkowymi. Po zakupieniu wersji spełniającej jego wymagania, nie może jednak skorzystać z programów dostosowanych do innych wersji systemu. Nie ma też prawie wyboru, gdy zakupi terminal, system przetwarzania tekstów lub system fotoskładu z już „wszczepionym" systemem operacyjnym.

Nie zawsze jest także korzystne, gdy jeden producent komputerów oferuje kilka systemów operacyjnych, ponieważ muszą one być dopasowane do komputera. Robi się to na ogół bez należytej staranności, co odbija się potem na użytkowniku, który cierpi wskutek ciągłych „załamań" systemu.

Dla użytkownika najważniejszą sprawą jest nabycie wypróbowanego i pewnego w działaniu systemu operacyjnego. Użytkownik powinien też uwzględnić możliwość przejścia na bardziej wydajny komputer. Musi też brać pod uwagę inne elementy oprogramowania systemowego, takie jak np. programy usługowe, służące m.in. do kopiowania i ochrony zbiorów danych. Ważne tu jest nie tylko bezpieczeństwo, ale i szybkość pracy.

micro

Oczami nauczycieli

Znamy już stanowisko Polskiego Towarzystwa Informatycznego w sprawie wprowadzania informatyki do szkół. Należy zapewne oczekiwać jeszcze kilku autorytatywnych wypowiedzi na ten temat. Jak jednak wygląda praktyka? Rozmowa, którą przeprowadziliśmy z nauczycielami ze środowiska warszawskiego, z pewnością nie jest prawdziwym obrazem rzeczywistości. W mniejszych ośrodkach zapewne jest jeszcze trudniej. Czy aby na pewno mierzymy siły na zamiary?

Do rozmowy o wprowadzaniu informatyki w szkołach zaprosiliśmy trzech nauczycieli: **Jerzego Kunickiego (J.K.)** ze Szkoły Podstawowej nr 278, **Henryka Daniszewskiego (H.D.)** z XXVIII LO im. J.Kochanowskiego oraz **Tadeusza Kurana (T.K.)** z XXVII LO im. T.Czackiego. W imieniu **MIKROKLANU (µk)** rozmawiał **Krzysztof Kontek.**

µk: – Czy w sytuacji, w jakiej obecnie znajdują się polskie szkoły, komputery i informatyka są im najbardziej potrzebne?

T.K.: – *Ja sądzę, że komputery należy kupować, nawet kosztem innych pomocy. Musimy wychować pokolenie przygotowane do życia w cywilizacji informatycznej. Muszą powstawać nowe programy i zastosowania. Jesteśmy w tej dziedzinie bardzo opóźnieni i dlatego musimy przeznaczać na rozwój informatyki duże środki. Trudno, piszmy w szkołach złą kredą. Zresztą, jeśli nie będzie komputerów, to nie widzę powodu, by kreda miała być lepsza. Jeżeli zrezygnujemy z komputerów, wcale nie będzie więcej ławek i książek. To są zupełnie niezależne problemy. Wiem z Ministerstwa Oświaty, że w skali roku pozostają niewykorzystane srodki na zakup pomocy naukowych rzędu miliardów złotych.*

H.D.: – *Przeciętnie sytuowanego ucznia długo jeszcze nie będzie stać na kupienie komputera, więc szkoła musi zapewnić mu dostęp do tego sprzętu.*

J.K.: – *Komputery nie są ważniejsze od innych pomocy naukowych, ale jeżeli są dostępne, to należy je kupować. Książek i tak więcej nie będzie ze względu na brak papieru. Ja nie sądzę, że zabieramy komus pieniądze. Szkoły borykają się rzeczywiście z wieloma problemami. W mojej szkole nie ma ławek dla starszych uczniów, prowadzi to do skrzywień kręgosłupa. Gdybym miał wybierac między ławkami i komputerami, wybrałbym ławki. Obecnie jednak komputery uzyskujemy głównie z darów.*

H.D.: – *Na wprowadzenie informatyki do szkół można spojrzeć także pod innym kątem. Należy uważac, by w Polsce za dwa, trzy lata nie stało się to, co na Zachodzie, gdzie zaobserwowano u młouzieży symptomy „choroby komputerowej". Siedem godzin w szkole, religia, angielski, klub filmowy i do tego informatyka spowodują, że wyrośnie chore społeczeństwo. Na miejscu dyrektora w szkole, w której nie działa SKS, nie pozwoliłbym na zajęcia kółka komputerowego. W zajęciach tych mogliby brać udział tylko ci uczniowie, którzy poświęcają co najmniej tyle samo czasu na ćwiczenia sportowe. Obowiązkowe zajęcia z informatyki lub udział w kółku komputerowym spowodują jeszcze większe obciążenie uczniów.*

µk.: – Jaki sprzęt macie w swoich szkołach i jakie jest jego źródło?

J.K.: – *Szkoła Podstawowa nr 278 ma komputer ZX Spectrum 16 KB, który dostaliśmy z darów, innego sprzętu nie mamy. Telewizor przynoszę z domu, magnetofon pożyczył uczeń. Kable musiałem przerobić sam.*

H.D.: – *W XXVIII LO mamy trzy komputery ZX Spectrum Plus, ofiarowane przez Polonusa z Anglii. Komitet Rodzicielski kupił jeden monitor czarno-biały, dwa monitory załatwia Wydział Oświaty. Magnetofony przynoszą uczniowie. Drukarkę mamy otrzymać z darów.*

T.K.: – *W Liceum im. Czackiego sytuacja jest najlepsza. Komputery ZX Spectrum zdobywaliśmy z różnych źródeł: z Polbritu, z darów, komisu, od szkoły podstawowej, w której nie wiedziano, co z nim zrobić, od PTI, Wydziału Oświaty oraz Mery (7 kompletnych zestawow Meritum). Mamy też jedną drukarkę D 100.*

µk.: – Ile szkół w ogóle nie ma komputerów?

H.D.: – *W szkołach podstawowych komputerów jest bardzo mało. Natomiast na terenie Warszawy, zdaje się, 12 liceów ma co najmniej po jednym komputerze. Niestety nie dysponuję dokładnymi danymi. Dobrze wyposażonych szkół jest zaledwie kilka*

µk.: – Jeżeli komputerów jest ciągle tak mało, to czy lepiej gromadzić je w wybranych szkołach czy też rozdzielać równo?

J.K.: – *Wydaje mi się, że należy umożliwić do-*

stęp do komputera każdemu dziecku.

H.D.: – *Ja natomiast uważam, że tych środków nie można rozpraszać. Mogę oddać swoje trzy komputery do innej szkoły. Komputery powinny trafiać tam, gdzie będą najlepiej wykorzystywane. Gdy do szkoły trafia pierwszy komputer, jest on zazwyczaj przez pół roku nie używany, bo brakuje monitora, magnetofonu, kabelka, programów. W szkole odpowiednio przygotowanej od razu znajduje on swoje zastosowanie.*

µk.: – Komputerów jest ciągle mało. Skąd można oczekiwać pomocy?

H.D.: – *Doraźnie można liczyć na pomoc rodziców. Rodzice chętnie dają pieniądze na określony sprzęt np. monitor. Znacznie chętniej niż na komitet rodzicielski. Natomiast na to, by rodzice kupili cały zestaw komputerowy nie ma co liczyć, bo jest on po prostu za drogi.*

T.K.: – *We Wrocławiu utworzono fundację, która gromadzi pieniądze wpłacane przez zakłady pracy z przeznaczeniem ich na sprzęt komputerowy dla szkół. O utworzeniu podobnych fundacji można byłoby pomyśleć i w innych częściach kraju. Ta forma pomocy dla szkoł jest korzystna dla zakładów pracy, gdyż pieniądze, przeznaczone na ten cel, nie są obciążone przez FAZ. Podobne fundacje mogliby założyć Polacy mieszkający za granicą i obcokrajowcy. Sprzęt kupowany za dolary jest jednak znacznie tanszy od oferowanego w kraju.*

H.D.: – *Myślę, że z taką propozycją mogłaby wystąpić Oświata na przykład do Towarzystwa Polonia. Jestem pewien, że osób chcących pomóc polskiej młodzieży byłoby dużo.*

µk.: – Należy także dążyć do tego, by cena sprzętu komputerowego przeznaczonego dla szkół maksymalnie zmalała. W ubiegłym roku zniesiono cło za komputery osobiste przywożone do kraju. Być może Ministerstwo Finansów mogłoby pójść jeszcze dalej i objąć ulgami podatkowymi producentów sprzętu komputerowego przeznaczonego dla szkół. Powinno to doprowadzić do obniżenia ceny sprzętu i zwiększenia jego produkcji. Należy także ułatwić zakup komputerów od osób prywatnych oraz obniżyć podatek pobierany przy tego typu transakcjach. Wydaje się, że takie decyzje mogłyby wpłynąć na szybki wzrost ilości sprzętu komputerowego w szkołach.
Jak wykorzystane są komputery w Waszych szkołach?

T.K.: – *W mojej szkole komputery są prawie cały czas wykorzystywane. Mamy przecież zajęcia z informatyki. Po południu, aż do godziny 20[00] i w soboty, odbywają się zajęcia kołek, na których uczymy języków Logo, BASIC oraz Asembler i to zarówno uczniów naszej szkoły, jak i innych, w tym 80 uczniów szkół podstawowych. Dzień pracy naszych komputerów trwa więc od 8[00] do 20[00].*

H.D.: – *U nas dzieci przychodzą rano i proszą, by mogły trochę popracować. W ten sposób 15 uczniow wykorzystuje komputer przez cały tydzień w czasie okienek, przerw i wagarów. Oczywiście ma to charakter nieformalny. W soboty prowadzę kółko, które zamiast planowych trzech godzin, trwa zwykle pięć.*

J.K.: – *Ja na razie stosuję komputer do użytku własnego. Staram się, by kółko rozpoczęło działalność w najbliższym czasie.*

µk.: – Jak wygląda sprawa oprogramowania, podręczników i instrukcji obsługi komputerów?

T.K.: – *Brakuje dobrych opisów języków programowania. Staram się o to, żeby wszystkie książki, które mogą przydać się na zajęciach, były u nas w szkole. Mamy wiele pozycji WNT z serii Biblioteka Inżynierii Oprogramowania. Opisy do Meritum są bardzo złe. Do Spectrum wykorzystujemy kopie odbijane na ksero. Uważam, że posiadanie dużej ilości literatury jest potrzebne. Jeśli ktoś chce pisać niewielkie programy edukacyjne, które same by uczyły i kontrolowały ucznia, w zasadzie może to zrobić korzystając tylko z tego, co jest napisane w instrukcji. Lepiej jednak poznać efektywniejsze metody. Na razie wiele materiałów odbijamy po prostu na ksero. Nie jest to oczywiście metoda na dłuższą metę. Ten nurt powinien być wspierany polskimi publikacjami.*

J.K.: – *Na razie mam tylko podręcznik do Spectrum w języku niemieckim oraz tłumaczenie na język polski. W tej chwili to mi wystarcza.*

H.D.: – *Podręczniki do programowania nie są mi na razie specjalnie potrzebne. Jestem pod tym względem minimalistą. Potrzebuję natomiast dobrych, prostych instrukcji do komputerów.*

µk.: – Wydaje się zatem, że w obecnej chwili najważniejsze byłoby dla Was wydanie dobrze przetłumaczonego opisu do komputera ZX Spectrum?

T.K.: – *Tak, i to opisu do ZX Spectrum Plus, który mimo że niezbyt obszerny, jest bardzo przejrzysty. Więcej jest tam rzeczowych informacji. Poza tym będziemy potrzebowali coraz więcej informacji o Meritum.*

J.K.: – *Innym problemem jest brak oprogramowania. Programy, które mam, napisałem sam lub też zdobyłem od kolegów.*

T.K.: – *W naszej szkole też powstało już kilka programów. Chętnie udostępnimy je innym szkołom. Wiem, że problemem oprogramowania zaczęto się już zajmować: powstała komórka przy Instytucie Badan Pedagogicznych, która będzie rozpowszechniać programy edukacyjne dla szkół. Został nawiązany także kontakt z firmą Preisler. Pierwszy program z fizyki o nazwie „Kinematyka" jest już sprzedawany, w cenie 2500 zł wraz z opisem. Problemem oprogramowania zajął się również Instytut Maszyn Matematycznych na zlecenie Mery. Także w programie opracowywanym przez Ministerstwo Oświaty przewiduje się utworzenie odpowiedniej komórki.*

H.D.: – *Programy edukacyjne powinny być kupowane od nauczycieli (być może także od uczniów) po takiej cenie, aby opłaciło się pisanie dobrych programow. Napisanie dobrego programu edukacyjnego nie jest bowiem proste. Wyobrażam sobie, że w przyszłosci programy będą traktowane jako zwykłe pomoce naukowe i będą rozprowadzane przez instytucje za to odpowiedzialne.*

µk.: – Czy poza sprawami sprzętu, oprogramowania i podręczników istnieją jeszcze jakieś bariery przed upowszechnieniem komputerów w szkołach?

T.K.: – *Bardzo poważną barierą może być znalezienie ludzi, którzy będą chcieli się tym zajmować. Nauczycieli w ogóle brakuje, a ci którzy są, zazwyczaj nie chcą uczyć się informatyki, mieć problemy z kabelkami. Informatycy do szkół na ogół nie przychodzą, bo nie zadowalają ich własne pensje, a cóż dopiero nauczycielskie.*

H.D.: – *Zwykle nauczyciel woli wziąć godzinę korepetycji za 500 zł niż godzinę kółek za 150 zł.*

J.K.: – *Będę prowadził kółko, bo to lubię, a nie dlatego, że mi się to będzie opłacało.*

T.K.: – *Część uczniów bardzo szybko zaznajamia się z komputerami, co powoduje powstanie u nauczycieli lęku przed utratą autorytetu. Z naszych doświadczeń wynika, że uczniowie często orientują się lepiej od nauczycieli i musimy się z tym pogodzić. Na pewno jednak nauczyciele, którzy przychodzą do szkół, muszą stykać się z mikrokomputerami już na studiach.*

H.D.: – *Jednym ze sposobów łamania barier psychicznych jest organizowanie kursów. Jednak bardzo często na kursy takie trafiają osoby zupełnie przypadkowe, wytypowane przez dyrektora np. dlatego, że nie mają dzieci.*

µk.: – Czy w nauczycielach nie powstaje obawa, że programy edukacyjne mogą ich zastąpić.

H.D.: – *Kredy i tablicy nie da się zastąpić.*

T.K.: – *Na pewno komputery nie wyeliminują nauczycieli, ale częściowo mogą realizować zadania, które do tej pory były obowiązkiem nauczyciela. Komputer pozwala przecież w wielkim stopniu indywidualizować nauczanie. Możliwe jest wykorzystywanie mikrokomputerów do testowania wiadomości uczniów. Pozwala to nauczycielom zaoszczędzić czas poświęcony dotychczas na odpytywanie uczniów i poświęcić go na dokładniejsze wyjaśnienie nowego materiału. Uczniowie zaś byliby oceniani w sposób obiektywny.*

µk.: – Jaką rolę komputer spełnia obecnie w

Waszych szkołach, jakie zadania są jeszcze przed nim?

H.D.: – *Na dzisiaj jest to po prostu znakomity magnes. Podczas gdy na zajęcia innych kółek młodzież nie przychodzi, ja muszę wyciągać wtyczkę z gniazdka, żeby wyjść ze szkoły. Młodzież jest głównie zainteresowana grami, chociaż niektórzy już tę fazę przeszli. Można sobie jednak wymarzyć, że w przyszłości biolog, geograf, chemik i inni nauczyciele będą mieli bogaty zbiór programów edukacyjnych i będą z nich korzystać na lekcjach. Na przykład, przy omawianiu praw Mendla uczniowie będą mogli zobaczyć, jak wyglądają kolejne pokolenia królików.*

Powinny odbywać się także zajęcia z informatyki. Uczenie programowania ma sens. Uruchomienie swojego, nawet bardzo prostego programu daje uczniowi ogromną satysfakcję.

T.K.: – *Młodzież z mojej szkoły chce napisać program do tworzenia rozkładu zajęć. Inny program ma pomagać przy sprzedaży podręczników. W najbliższym czasie będziemy mieli program do planowania zastępstw.*

Komputer może również odegrać poważną rolę w edukacji. Obecnie prowadzimy naukę programowania w języku BASIC i Logo, bo chcemy uczyć algorytmicznego sposobu rozumowania i rozwiązywania problemów. Taki sposób myślenia jest ważny, gdyż uczy dokładności i ścisłości. Poza tym komputery mogą pomóc w kształtowaniu wyobraźni i rozwijaniu najogólniej rozumianych możliwości twórczych.

Z wypowiedzi tych wynika, że sytuacja w polskim szkolnictwie, poza kilkoma szkołami, takimi jak Liceum im. Czackiego, nie jest najlepsza. Wiele szkół nie ma żadnego komputera, część szkół posiada pojedyncze egzemplarze. Zdarza się, że nawet i te nie są wykorzystywane – czasami z braku monitora lub magnetofonu, niekiedy z braku chęci do prowadzenia zajęć. Tylko nieliczne szkoły mają dobrze wyposażone pracownie. W większości przypadków komputery są dostarczane z darów, a nie przez przemysł krajowy. Wszystkie problemy związane z prowadzeniem kółek spadają na nauczycieli, którzy przynoszą własne telewizory i magnetofony, przerabiają kabelki, programy zdobywają od kolegów albo piszą je sami, a instrukcje odbijają na kserografach. Nie istnieją przy tym żadne mechanizmy materialnej zachęty dla tych osób. Wydaje się, że wprowadzenie informatyki w szkołach jest zbyt poważnym zadaniem, by opierać je tylko na zapale i improwizacji nielicznych nauczycieli.

KRZYSZTOF KONTEK

Pamięć zewnętrzna mikrokomputerów na minidyskach elastycznych (1)

Każdy, kto bliżej zetknął się z systemem mikrokomputerowym z pewnością zgodzi się, że z punktu widzenia użytkownika, wyjątkowo istotną cechą takiego systemu jest niezawodna i szybka pamięć zewnętrzna. Trwały, niezależny od zasilania, zewnętrzny nośnik informacji jest niezbędny do przechowywania programów, tekstów i danych oraz pozwala na wprowadzenie do pamięci RAM podstawowego oprogramowania (systemu operacyjnego, translatora języka wysokiego poziomu itp.). Inne parametry systemu, takie jak rozmiar pamięci operacyjnej, długość słowa, typ i szybkość procesora są niewątpliwie również istotne, lecz komputer nie posiadający dobrych możliwości wprowadzenia i zapamiętywania rezultatów pracy człowieka jest mało przydatny. Warto zainwestować w pamięć zewnętrzną dobrej jakosci, by zabezpieczyć się przed możliwością utraty wyników wielu godzin pracy.

Historia systemów mikrokomputerowych zna wiele zewnętrznych nosnikow informacji. Należały do nich, jeszcze niedawno, taśmy dziurkowane, które ze względu na małą pojemność, duży rozmiar, koszt i hałaśliwość urządzeń wejścia-wyjścia stopniowo odchodzą już w zapomnienie. Proste komputery osobiste wykorzystują najczęściej kasety magnetofonowe. Niestety, cechuje je mała prędkość transmisji i stosunkowo duża zawodność.

Dane umożliwiające porównanie wybranych cech niektórych typów pamięci zewnętrznych przedstawiono w tabeli 1.

Obecnie nośnikiem, który pod względem niezawodności i szybkości działania, a jednocześnie wygody, funkcjonalności i ceny, może zadowolić nawet kapryśnych użytkowników sprzętu mikrokomputerowego, jest dobrej jakości minidyskietka o średnicy 5,25".

Większość profesjonalnych systemów mikrokomputerowych jest wyposażona w jeden (lub kilka) napędów (stacji) dyskow elastycznych. Mimo to, a może właśnie dlatego, pragniemy przybliżyć w kilku kolejnych artykułach sposób podłączenia i realizacji współpracy komputera z dyskietkami 5.25".

Sposób magazynowania informacji na minidyskietkach przechodzi szybką ewolucję. Doprowadziło to do stworzenia wielu standardów i formatów zapisu. Do najczęściej spotykanych różnic należy: zapis po jednej stronie (SS – ang. Single Side) lub po obu stronach dysku (DS – ang. Double Side) oraz zapis pojedynczej (SD – ang. Single Density) lub podwójnej gęstości (DD – ang. Double Density). Ponadto, dzięki poprawie jakości nośnika magnetycznego zagęszczeniu ulegają ścieżki, na których jest dokonywany zapis. Podstawowe parametry różnych formatów zapisu na minidyskach elastycznych przedstawiono w tabeli 2.

Komunikacja pomiędzy procesorem a jednostką (napędem) dyskietek odbywa się za pośrednictwem bloku funkcjonalnego, nazywanego sterownikiem dysków. Z reguły jego budowa oparta jest na specjalizowanym układzie o wielkim stopniu scalenia (LSI).

Tabela 1. Właściwości wybranych rodzajów pamięci zewnętrznych

Cecha	Taśma papierowa	Kaseta magnetofonowa	Dysk elastyczny
Pojemnośc	16 KB	100 KB	300 KB
Czas przygotowania	3 min.	40 s	40 s
Czas dostępu	▮ min.	40 s	10 s
Czas czytania 4 KB	15 s	1 min.	5 s
Czas zapisu ▮ KB	▮ min.	1 min.	5 s
Odpornośc mechaniczna	mała	duża	mała
Odpornośc magnetyczna	b. duża	mała	mała
Możliwośc usunięcia uszkodzeń zbiorów	duża	b. mała	mała
Częstośc błędów	1×10^{-4}	1×10^{-5}	1×10^{-6}
Wygoda tworzenia archiwum zbiorów	b. mała	mała	duża

Przykładowym realizacjom takiego sterownika poświęcimy następne artykuły. Przyjrzyjmy się teraz dalszej części tego toru: interfejsowi, obejmującemu sygnały łączące sterownik z napędem, oraz funkcjom realizowanym przez samą jednostkę dyskową.

Jednostki lub napędy (ang. drive) dysków elastycznych, oferowane przez renomowane firmy amerykańskie i japońskie, są wyposażone w standardowe złącze (zarówno pod względem logicznym, jak i fizycznym). Do tego standardu próbują dostosować się mniejsze firmy (również w krajach RWPG) – z różnym skutkiem. Zestaw sygnałów takiego złącza przedstawiono w tabeli 3. Interfejs obejmuje 11 sygnałów wyjściowych, wychodzących z komputera do jednostki dyskowej, oraz 5 sygnałów wejściowych, prowadzących z napędu do komputera. Sposób przyłączenia dyskietek 5,25" do mikrokomputera przedstawiono schematycznie na rysunku 1.

Sygnały wyjściowe realizują następujące funkcje:

★ **MOTOR ON** – włączenie silnika obracającego dyskietką z szybkością 6 obr/s. Sygnał włącza silnik dysku niezależnie od stanu innych linii, ale nie powoduje dołączenia sygnałów wejściowych interfejsu.

★ **DRIVE SELECT** (0,..,3) – wybór jednej z czterech jednostek dyskowych, dołączonych do równoległej magistrali. Każdy sygnał powoduje dołączenie jednej z czterech jednostek do magistrali, sygnalizując jednocześnie, że pozostałe sygnały interfejsu są ważne i przeznaczone dla wybranego dysku. Sygnał **DRIVE SELECT** niezależnie od **MOTOR ON** może włączać silnik napędzający dyskietkę. Sposób sterowania włączeniem silnika za pomocą synałów **MOTOR ON** i **DRIVE SELECT** zależy od połączeń wykonanych wewnątrz jednostki.

★ **DIRECTION SELECT** – określa kierunek przesuwania głowicy odczytująco-zapisującej (do środka lub od środka dysku).

★ **STEP** – każdy impuls na tej linii sygnałowej powoduje przesunięcie głowicy dysku wybranego przez sygnał **DRIVE SELECT** nad następną ścieżkę. Kierunek przesunięcia określa sygnał **DIRECTION SELECT**.

★ **HEAD LOAD** – powoduje opuszczenie głowicy odczytującej na powierzchnię wybranego dysku. Sygnał ten w połączeniu z sygnałem **DRIVE SELECT** ma wpływ na włączenie się lampki określającej pracę dysku.

★ **SIDE ONE SELECT** – pozwala na dostęp do drugiej strony wybranego dysku, w przypadku współpracy z dyskiem zapisywanym dwustronnie (jednostka dyskowa musi mieć dwie głowice).

★ **WRITE GATE** – umożliwia zapisywanie i kasowanie informacji na wybranym dysku, o ile sygnał **WRITE PROTECT** nie jest aktywny.

★ **WRITE DATA** – przebieg w zakodowanej formie, dostarczający zapisywane na dysku bity danych. Postać danych w przypadku zapisu z podwójną gęstością w systemie MFM przedstawiono na rysunku 2.

Sygnały wejściowe dostarczają następujących informacji:

★ **INDEX** – sygnalizacja fizycznego początku ścieżki, generowana przez czujnik fotoelektryczny, wykrywający pojawienie się otworu

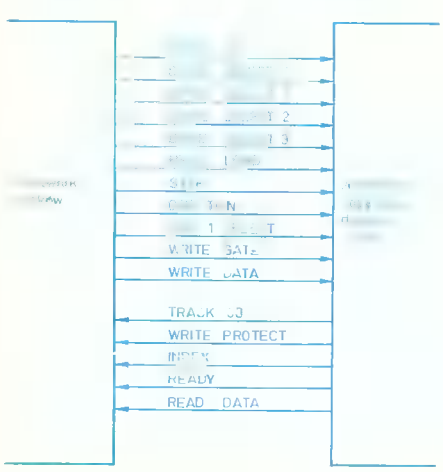
Tabela 2. Podstawowe parametry minidysków zapisanych po jednej (SS) i po obu stronach (DS), pojedynczej (SD) i podwójnej gęstości zapisu (DD) przy gęstości 48 i 96 ścieżek na cal (TPI)*

Typ dysku	SSSD	SSDD	DSSD	DSDD	SSSD	SSDD	DSSD	DSDD
Gęstość ścieżek (TPI)	48	48	48	48	96	96	96	96
Liczba ścieżek na dysku	40	40	80	80	80	80	160	180
Szybkośc obrotowa (obr/min)	300	300	300	300	300	300	300	300
Czas zmiany ścieżki (ms)	6	6	6	6	3	3	3	3
Średni czas dostępu (ms)	93	93	93	93	93	93	93	93
Szybkośc transmisji (Kb/s)	125	250	125	250	125	250	125	250
Pojemność:								
niesformatowana								
– ścieżki (KB)	3,12	6,25	3,12	6,25	3,12	6,25	3,12	6,25
– dysku (KB)	125	250	250	500	250	500	500	1000
sformatowana								
– ścieżki (KB)	2	4	2	4	2	4	2	4
– dysku (KB)	80	160	160	320	180	320	320	640
Pojemność w przeliczeniu na maszynopis (strona = 60x30 znaków)	45	91	91	182	91	182	182	364

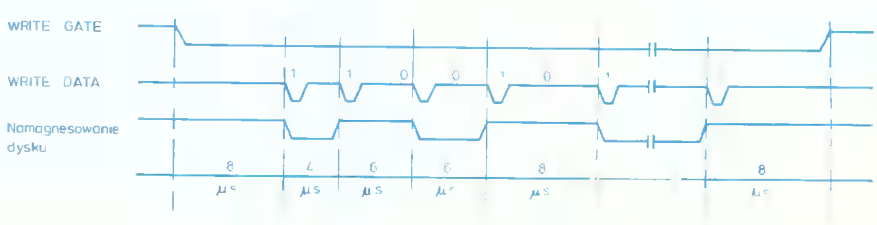
* TPI – ang. track per inch – ścieżek na cal

Tabela 3. Zestaw sygnałów łącza jednostki dyskowej ze sterownikiem

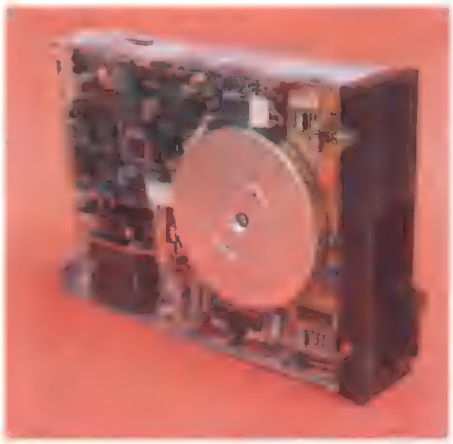
Nazwa sygnału	Kierunek	Numer końcówki złącza	
		sygnał	masa
	od komputera	2	1
HEAD LOAD	od komputera	4	3
DRIVE SELECT 3	od komputera	6	5
INDEX	do komputera	8	7
DRIVE SELECT 0	od komputera	10	9
DRIVE SELECT 1	od komputera	12	11
DRIVE SELECT 2	od komputera	14	13
MOTOR ON	od komputera	16	15
DIRECTION SELECT	od komputera	18	17
STEP	od komputera	20	19
WRITE DATA	od komputera	22	21
WRITE GATE	od komputera	24	23
TRACK 00	do komputera	26	25
WRITE PROTECT	do komputera	28	27
READ DATA	do komputera	30	29
SIDE ONE SELECT	od komputera	32	31
READY	do komputera	34	33



Rys. 1. Przyłączenia dysków 5,25" do mikrokomputera



Rys. 2. Postać danych w przypadku zapisu z podwójną gęstością w kodzie MFM



Napęd dysków 5,25" (TEAC FD-55B)

w powierzchni obracającej się dyskietki. Otwór ten sygnalizuje, że od tego miejsca rozpoczyna się zapisywanie sektorow na ścieżce.

★ **TRACK OO** – sygnał generowany przez mechaniczny mikrowyłącznik, określający położenie głowicy odczytująco-zapisującej nad zewnętrzną ścieżką dysku. Sygnał pozwala sprowadzić głowicę do pozycji początkowej.

★ **WRITE PROTECT** – informuje o ochronie dysku przed zapisem. Jest generowany przez czujnik wykrywający fizyczne zamknięcie otworu ochrony przed zapisem, jaki znajduje się na bocznej krawędzi dysku.

★ **READ DATA** – sygnał impulsowy, o postaci zbliżonej do sygnału **WRITE DATA**, generowany na podstawie przebiegu odczytanego z dysku.

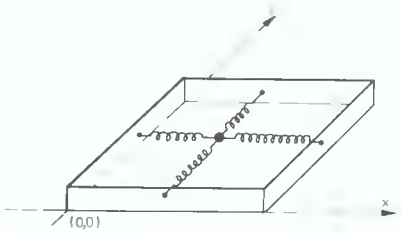
Na podstawie opisu sygnałów łącza napędu dyskow z komputerem można wyciągnąć wniosek, że jednostka dyskowa nie jest urządzeniem obdarzonym dużą inteligencją. Sygnały dostarczane do niej mają bardzo prostą postac i ulegają tylko nieznacznym przetworzeniom (z postaci cyfrowej na analogową). Cała inteligencja, pozwalająca na przesyłanie bajtów danych z szybkością kilkuset Kbitów na sekundę oraz zapewniająca dostęp do wybranego sektora, średnio w czasie 1/10 sekundy, znajduje się w jednostce sterującej napędem dysku.

cdn.

ADAM PLUTA

Symulacja drgań w dwóch wymiarach

Komputer można zastosować do symulacji aparatury służącej do przeprowadzenia ekperymentów naukowych. Takie podejście ma wiele zalet: przed podjęciem właściwych badań można oszacować potrzebne środki, zakres poszukiwań i spodziewane rezultaty. Czasami po symulacji eksperymenty praktyczne okazują się w ogóle zbędne. Symulacja jest tania, szybka i bezpieczna.



Rys. 1. Model fizyczny punktu drgającego w dwóch wymiarach

Przedstawiamy program, który wykreśla na ekranie telewizora trajektorię punktu drgającego jednocześnie w dwóch prostopadłych do siebie kierunkach. Modelem fizycznym może tu być kula, zamocowana za pomocą czterech sprężyn do nieruchomej obudowy, wyznaczającej układ współrzędnych (rys. 1.). W najprostszym przypadku kula porusza się po elipsie (w szczególnym przypadku po okręgu lub odcinku), wykonując równoczesne drgania w kierunkach osi *x* i osi *y*. Ze względu na opory ruchu, drgania te gasną z upływem czasu – trajektoria zatem nie jest krzywą zamkniętą, lecz raczej spiralą zwijającą się do środka. Ruch ten można opisać, podając zależność od czasu współrzędnych *x* i *y* drgającego punktu:

$$x(t) = X \sin(\omega t + \varphi_x) e^{\frac{-t}{sd}x}$$

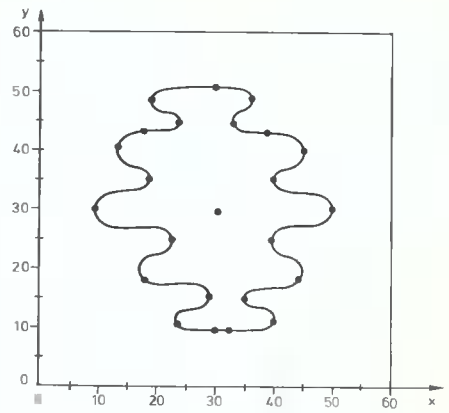
$$y(t) = Y \sin(\omega t + \varphi_y) e^{\frac{-t}{sd}y}$$

Za tłumienie jest odpowiedzialny człon $e^{\frac{-t}{sd}}$

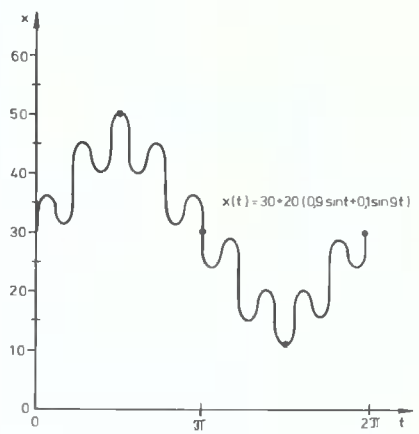
Ruchy w każdym z kierunkow nie muszą być opisane zwykłymi gasnącymi sinusoidami, lecz mogą być złożeniem wielu drgań. Można je wtedy opisać następującymi równaniami:

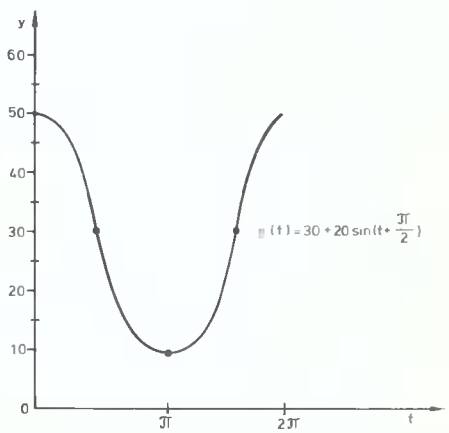
$$x(t) = \sum_{j=0}^{n} X_j \sin(\omega_{xj} + \varphi_{xj}) e^{\frac{-t}{sd}x}$$

$$y(t) = \sum_{j=0}^{n} Y_j \sin(\omega_{yj} + \varphi_{yj}) e^{\frac{-t}{sd}y}$$



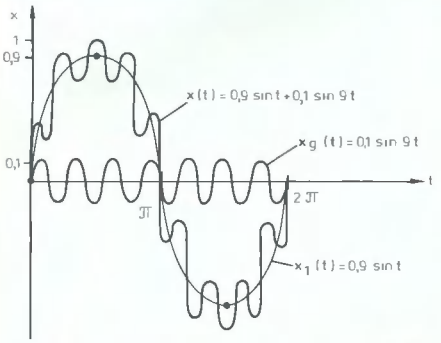
Rys. 2. Trajektoria w kształcie okręgu zniekształconego poziomymi pasami



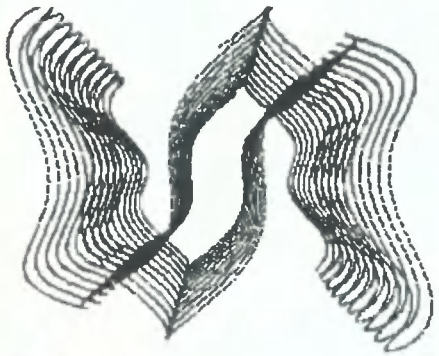


Rys. 3. Zależności współrzędnych x i y

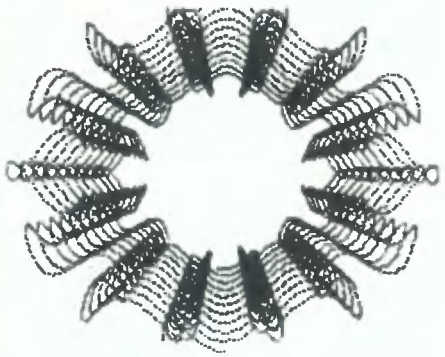
```
  1 REM ****SWING'O'GRAPH****
  2 REM BERNARD PIOTR ZURAWSKI  GDYNIA 1985
  3 REM ================================
  4 REM
 20 BORDER 0: PAPER 6: INK 7
 30 CLEAR : CLS
 40 DIM a(30,3,2): DIM b(2): DIM c(2): DIM d(2): DIM e(2)
 50 PRINT #1;"Czy zaladowac obraz z tasmy?    (T/N)"
 60 IF INKEY$="t" OR INKEY$="T" THEN  GO TO 1100
 70 IF INKEY$="n" OR INKEY$="N" THEN  GO TO 90
 80 GO TO 60
 83 REM
 84 REM ====================
 85 REM ****Wczytanie danych***
 86 REM ====================
 87 REM
 90 FOR i=1 TO 2
100 CLS : INPUT "Ile przebiegow do zsumowania na osi (maks. 30) ";i;
" ? ";b(i): IF b(i)>30 THEN  PRINT "Sumuje tylko do 30 przebiegow"'"Pro
sze podac ponownie": PAUSE 150: GO TO 100
110 FOR j=1 TO b(i)
120 INPUT "Podaj numer harmonicznej skladowej ";(j);" przebiegu ";(i);
" ? ";a(j,1,i)
130 INPUT "Podaj amplitude skladowej ";(j);" przebiegu ";(i);" ? ";a(j
,2,i)
140 INPUT "Podaj przesuniecie fazowe w stopniach skladowej ";(j);" prz
ebiegu ";(i);" ? ";a(j,3,i)
150 NEXT j
160 NEXT i
170 PRINT "Informacje o wykresie": INPUT "Dlugosc wykresu ? ";wm
180 INPUT "Dokladnosc wykresu ?"'"( srednio 0.005 do 0.01 ) ";ws
190 INPUT "Amplituda pozioma ?"'"( maks. 127 )";rx: IF rx>127 TH
EN  GO SUB 220: GO TO 190
200 CLS : INPUT "Amplituda pionowa "                ( maks. 87 ) ";ry: IF
ry>87 THEN  GO SUB 220: GO TO 200
210 GO TO 240
220 PRINT AT 0,0;"Za duza amplituda"'"Prosze podac ponownie": PAUSE 10
0
230 RETURN
240 INPUT "Tlumienie ? ( normalnie i00 ) ";sd
'

243 REM
244 REM ====================
245 REM Kontrolny wydruk danych
246 REM ====================
247 REM
250 CLS : PRINT AT 0,0;"%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%"
260 FOR n=1 TO 3
270 PRINT AT n,0;"%";AT n,31;"%"
280 NEXT n
290 PRINT AT 2,4;"S P R A W D Z   D A N E !"
300 PRINT AT 4,0;"%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%"
310 PRINT
320 PRINT "    P-Numer przebiegu"'"      odksztalconego"
330 PRINT "    S-Numer skladowej przebiegu"
340 PRINT "   NH-Numer harmonicznej"
350 PRINT "   Am-Amplituda"
360 PRINT "Przes-Przesuniecie fazowe"
370 PRINT : PRINT "P S  NH      Am          Przes."
380 FOR i=1 TO 2
390 FOR j=1 TO b(i)
400 PRINT TAB 0;i;" ";j;TAB 5;a(j,1,i);TAB 12;a(j,2,i);TAB 24;a(j,3,i)
410 NEXT j
420 NEXT i
430 PRINT : PRINT "Dlugosc maks. wykresu ";wm
440 PRINT "Dokladnosc wykresu ";ws
450 PRINT "Amplituda pozioma ";rx
460 PRINT "Amplituda pionowa ";ry
470 PRINT "Tlumienie ";sd
480 PRINT : PRINT
490 PRINT "Nacisnij    B-dane bledne                    Z-dane zgodne"
500 IF INKEY$="b" OR INKEY$="B" THEN  GO TO 530
510 IF INKEY$="z" OR INKEY$="Z" THEN  GO TO 750
520 GO TO 500
530 CLS
540 PRINT "Nacisnij :"
550 PRINT "P-Poprawienie danych przebiegu    skladowego"
560 PRINT "G-Poprawienie danych globalnych   wykresu"
570 PRINT "W-ponowne wyswietlenie danych"
580 PRINT "K-Koniec poprawiania"
590 IF INKEY$="w" OR INKEY$="W" THEN  GO TO 250
600 IF INKEY$="g" OR INKEY$="G" THEN  GO TO 170
610 IF INKEY$="k" OR INKEY$="K" THEN  GO TO 700
620 IF INKEY$="p" OR INKEY$="p" THEN  GO TO 640
630 GO TO 590
640 INPUT "Podaj numer przebiegu odksztalconego, w ktorym wystapil bla
d ? ";i
650 IF i>=1 AND i<=2 THEN  GO TO 680
660 PRINT AT 10,0;"Nie mozna podac przebiegu ";i;" skoro istnieja tylk
o 2": PAUSE 300
670 PRINT AT 10,0;"                              ": GO TO 640
680 INPUT "Podaj, w ktorym przebiegu skladowym przebiegu odksztalconeg
o ";(i);" wystapil blad ? ";j
690 IF j>=1 AND j<=b(i) THEN  GO TO 720
700 PRINT AT 10,0;"Nie mozna podac przebiegu skladowego ";j;" skoro w
przebiegu odksztalconym ";i;" zadeklarowano tylko ";b(i);" skladowych "
: PAUSE 250
710 PRINT AT 10,0;"                              ": GO TO 680
720 CLS
730 PRINT AT 0,0;"Stwierdziles w skladowej ";j;" przebiegu ";i;" blad.
Musisz teraz podac wspolczynniki Fourierra tej skladowej."
740 INPUT "Podaj numer harmonicznej skladowej ";(j);" przebiegu ";(i);
" ? ";a(j,1,i)
750 INPUT "Podaj amplitude skladowej ";(j);" przebiegu ";(i);" = ";a(j
,2,i)
760 INPUT "Podaj przesuniecie fazowe w stopniach skladowej ";(j);" prz
ebiegu ";(i);" ? ";a(j,3,i)
770 GO TO 530
780 CLS : PRINT AT 10,0;"Gdy bedziesz chcial zakonczyc"'"rysowanie wyk
resu wcisnij"'"klawisz  S"'''"Aby rozpoczac rysowanie"'"wykresu wcisni
j dowolny"'"klawisz"
790 PAUSE 100: PAUSE 0
793 REM
794 REM ====================
795 REM ****Czesc operacyjna***
796 REM ====================
797 REM
800 LET d(i)=0: LET d(2)=0
810 FOR i=1 TO 2
820 FOR j=1 TO b(i)
830 LET d(i)=d(i)+ABS a(j,2,i)
840 NEXT j
850 NEXT i
860 LET d(1)=rx/d(1)
870 LET d(2)=ry/d(2)
880 CLS
890 FOR w=0 TO 360 STEP ws
900 LET c(1)=0: LET c(2)=0
910 FOR i=1 TO 2
920 FOR j=i TO b(i)
930 LET c(i)=c(i)+a(j,2,i)*SIN (a(j,1,i)*w+a(j,3,i)*PI/180)
940 IF INKEY$="s" OR INKEY$="S" THEN  GO TO 1020
950 NEXT j
960 NEXT i
970 FOR i=1 TO 2
980 LET e(i)=d(i)*1/EXP (w/sd)
990 NEXT i
1000 PLOT 127+c(i)*e(i),87+c(2)*e(2)
1010 NEXT w
1020 PRINT #1;"Czy chcesz zaladowac obraz"'"na tasme ? Tak / Nie
1030 IF INKEY$="t" OR INKEY$="T" THEN  GO TO 1060
1040 IF INKEY$="n" OR INKEY$="N" THEN  GO TO 20
1050 GO TO 1030
1060 INPUT "Podaj nazwe obrazu ? ";a$
1070 SAVE a$SCREEN$ : SAVE a$ CODE a()
1080 PRINT AT 0,0;"Zaladowano"
1090 GO TO 20
1100 CLS : PRINT AT 0,0;"Ladowanie obrazu "
1110 LOAD ""SCREEN$
1120 PRINT #1;"CZY:Wydrukowac Zaladowac nowy"'"Opuscic obraz"
1130 IF INKEY$="o" OR INKEY$="O" THEN  GO TO 20
1140 IF INKEY$="z" OR INKEY$="Z" THEN  GO TO 1100
1150 IF INKEY$="w" OR INKEY$="W" THEN  COPY
1160 GO TO 1130
```



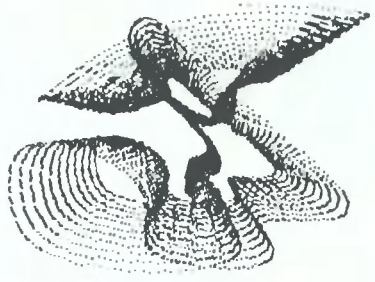
Rys. 4. Zmiany współrzędnej x jako złożenie dwóch drgań o różnych częstotliwościach



Swing 1



Swing 2



Swing 3

Przykładem (bez wprowadzenia tłumienia) może być ruch opisany wzorem:
x (t) = 30 + 20 (0,9 sin t + 0,1 sin 9t)
y (t) = 30 + 20 sin (t + π/2)

Trajektorię i zależności czasowe współrzędnych przedstawiono na rys. 2 i 3. Na rysunku 4 pokazano, że ruch w kierunku osi x jest złożeniem dwóch drgań: o częstotliwości podstawowej i o częstotliwości 9 razy większej.

Prawdziwe jest twierdzenie, że dobierając odpowiednie współczynniki X_i, ω_{xi}, φ_{xi} oraz Y_i, ω_{yi}, φ_{yi} i biorąc odpowiednio dużo składowych drgań bez tłumienia możemy przedstawić dowolny ruch okresowy. Człon $e^{\frac{-t}{sd}}$ może dodatkowo uwzględnić tłumienie. Poeksperymentujemy więc badając, jak w zależności od przyjętych współczynników, zmienia się trajektoria ruchu. Obliczenia i wykresy wykona mikrokomputer ZX Spectrum.

Program został napisany z myślą o wygodzie użytkownika. Wystarczy odpowiadać na pytanie zadawane przez komputer. Można więc szybko wprowadzać wyliczone parametry i obserwować, jak wpływają one na trajektorię ruchu.

Parametry globalne to:

ws – odstęp czasu między kolejnymi próbkowaniami (im mniejszy, tym dokładniejszy wykres, lecz dłuższy czas obliczeń);
wm – całkowity czas obserwacji
rx – amplituda drgań w kierunku *x; ry* - amplituda drgań w kierunku *y*
sd – parametr określający tłumienie.

Na podstawie podanych amplitud wykres zostanie tak wyskalowany, aby w całości zmieścił się na ekranie.

W wierszu 40 zdefiniowano używane w programie tablice. Przechowują one następujące wartości:

- a (30,3,2) – maksimum 30 grup współczynników opisujących poszczególne drgania składowe, zgrupowanych po 3 (x_i, ω_{xi}, φ_{xi}) lub (y_i, ω_{yi}, φ_{yi}) dla każdej z osi *x* i *y*;
- *b*(2) – liczba rozpatrywanych składowych dla każdej z osi *x* i *y*;
- *c*(2) – chwilowe wartości współrzędnych *x(t)* i *y(t)* bez skalowania i bez tłumienia dla każdej z osi *x* i *y*;
- *d*(2) – współczynnik skalowania dla każdej z osi *x* i *y*;
- e(2) – wartość członu tłumiącego $e^{\frac{-t}{sd}}$ dla każdej z osi *x* i *y*.

Wartości poszczególnych elementów tablic *a* i *b* są wczytywane z klawiatury, wartość zaś elementów tablic *c, d* i *e* wylicza się w programie.

Dane globalne i współczynniki po wprowadzeniu są wyświetlane na ekranie – można wtedy poprawić ewentualne błędy. Jeśli wszystko jest w porządku i użytkownik nie zgłasza chęci wprowadzenia poprawek, następuje przejście do obliczeń i wykreślenie trajektorii na ekranie. Wykres z ekranu może być skopiowany na drukarkę lub zapisany na taśmie magnetofonowej w celu późniejszego ponownego odtworzenia.

Zachęcamy do wprowadzenia przedstawionego programu i eksperymentów. Jak otrzymać następujące figury: punkt, odcinek, okrąg, elipsę, spiralę, kwadrat? A jak inne ciekawe wzory?

BERNARD PIOTR ŻURAWSKI

Apple II w sieci lokalnej

Firma Glanmire Electronics oferuje płytki, które po zainstalowaniu w komputerze Apple II pozwalają na utworzenie lokalnej sieci tych komputerów. Płytka zastępuje w systemie dotychczasowy sterownik napędu dyskietek i stanowi moduł wielofunkcyjny, obejmujący sterownik dysków, port dla drukarki oraz oczywiście sterownik dostępu do sieci. W takiej konfiguracji komputer może działać pod nadzorem systemów operacyjnych DOS, ProDOS i in. Cena płytki ok. 1800 franków francuskich.

TRASPUTER już na rynku

Doszły do nas informacje, że ukazał się już na rynku pierwszy model mikroprocesora Transputer IMS T414 (firma INMOS). Transputer, jeden z najbardziej złożonych układów scalonych, jakie powstały w historii półprzewodników, zawiera w sobie następujące bloki funkcjonalne: jednostkę centralną 32-bitową o mocy przetwarzania 10×10^6 instrukcji /s, 2 KB pamięci RAM, sterownik zewnętrznej pamięci dynamicznej i układy interfejsu. Układ, wykonany w technologii CMOS 1,5 µm, zawiera ponad 200 tys. tranzystorów.

Transputer jest układem o architekturze RISC (ang. Reduced Instruction Set Computer). Czas cyklu procesora 50 ns. Równocześnie z układem scalonym, firma udostępniła system uruchomieniowy wraz z oprogramowaniem.

Dyski typu Whitney

Z pewnością większość wtajemniczonych użytkowników różnych mutacji IBM PC marzy o zainstalowaniu w swoim systemie sztywnego dysku typu Winchester. Wszystkim tym, którzy z różnych względów nie dorobili się jeszcze własnych 10 czy 20 MB spieszymy donieść, że firma IBM opracowała już nową generację pamięci masowej. Nowa technologia, nazwana Whitney, pozwala na osiągnięcie jeszcze większych gęstości zapisu. Opanowanie tej technologii, zawierającej nowinki z zakresu mechaniki, rejestracji magnetycznej i sposobu kodowania, zaowocowało pojawieniem się na rynku pierwszego Whitney'a – najbardziej zaawansowanej, spośród produkowanych seryjnie, jednostki pamięci masowej. Nosi ona oznaczenie IBM 3380. Poniżej zestawione są niektóre parametry jednostek starego i nowego typu.

Warto dodać, że jednostki typu Whitney mają interfejs zgodny ze swoim poprzednikiem.

Parametry	Winchester IBM 3350	Whitney IBM 3380
Liczba bitów na cal	6 425	15 000
Liczba ścieżek na cal	480	800
Liczba bitów na cal²	$3{,}084 \cdot 10^6$	$12 \cdot 10^6$
Średni czas dostępu	25 ms	16 ms
Prędkość transmisji	1,198 MB/s	3 MB/s

CENY

Pragniemy regularnie informować Czytelników o możliwie aktualnych cenach podzespołów, programów i sprzętu mikrokomputerowego (w tym również niedostępnego dla polskich użytkowników). Ceny zagraniczne podawane będą w walucie danego kraju (dolary, marki, funty), natomiast krajowe – na podstawie notowań „perskiego rynku" w Warszawie (obecnie w pobliżu skrzyżowania ulicy Obozowej i Ks. Janusza, w niedziele, godziny 7 – 12). Ceny krajowe sprzętu komputerowego podawane będą w tys. zł, a zestawienia – oprócz notowań wolnorynkowych – uwzględniać będą również ceny stosowane przez przedsiębiorstwa państwowe: BOMIS – B oraz PEWEX – P (w tym ostatnim przypadku oczywiście w dolarach). Przy okazji informujemy, że przedsiębiorstwo BOMIS zajmuje się obecnie również skupem i odsprzedażą zagranicznego sprzętu komputerowego.

Ceny z czerwca 1986

	Wielka Brytania (funty)	RFN (marki)
	Komputery	
Spectrum Plus	96	329
Sinclair QL	160	598
Commodore CBM 64	150	470
Commodore CBM 128	226	778
Commodore CBM 128 + CBM 1570 (napęd dysków)	426	1650
Amstrad CPC 6128 (monochromatyczny monitor)	252	1450
Amstrad CPC 6128 (kolorowy monitor)	339	1900
Apple IIc (64 KB + napęd dysków)	700	1995
Atari 520 ST (monitor + napęd dysków)	730	2680
Apricot F1 (256 KB + naped dysków 720 KB)	699	2598
IBM PC (256 KB + napęd dysków 360 KB)	950	1900
Olivetti M24 (128 KB + napęd dysków 360 KB)	–	6790
Apricot PC (256 KB + 2 napędy dysków)	1150	–
Ferranti PC 860 (256 KB + 2 napędy dysków)	1250	–
IBM PC – XT (256 KB + 2 napędy dysków 360 KB)	899	2100
IBM PC – XT (256 KB + 1 napęd 360 KB + sztywny dysk 10 MB)	1970	4250
	Drukarki mozaikowe	
Shinua CPA 80	170	548
Seikosha SP 1000	200	–
Epson Lx 80	200	795
Star SG 10	200	760
Canon 1080	270	–
	Monitory	
Philips BM 7502 (Monochromatyczny)	80	160
Philips CM 8501 (RGB 14")	182	–
Microvitec 1451 (kolorowy)	230	–
	Napędy dysków elastycznych	
5 1/4" 48/96 TPI	70...95	360
Amstrad DDI (z jednostką sterującą)	140	–
Amstrad FDI	90	–
Commodore 1541	130	–
Commodore 1570	170	750
	Dyski sztywne	
10 MB Winchester	330...500	1300...1500
20 MB Winchester	450...650	1700...1900
	Dyski elastyczne (10 sztuk)	
8" SS-SD	12	65
8" SS-DD	15	80
8" DS-DD	17	90
5 1/4" SS DD	14	40
5 1/4" SS QD	18	54
5 1/4" DS DD	18	50
5 1/4" DS QD	23	64
3 1/2" SS DD	30	80
3 1/2" DS DD	34	110

JAK PISAĆ DO MIKROKLANU...

MIKROKLAN jest dla Ciebie. Oznacza to, że i od Ciebie zależy, co będzie w nim drukowane. Będziemy wdzięczni, jeżeli napiszesz dlaczego Ci się to czy tamto nie podoba, a szczególnie wdzięczni, gdy zaproponujesz co zrobić, żeby było lepiej. Pamiętaj jednak, że papier nie jest z gumy i mieści się na nim skończona liczba informacji. Weź też pod uwagę prawdopodobną opinię pozostałych 99 999 Czytelników (niektórzy spośród nich są bez mała ekspertami, inni zaś stawiają pierwsze kroki...). Nadsyłane nam propozycje tematów, które zdaniem Czytelników powinny znaleźć się na łamach MIKROKLANU, kwalifikować będziemy według liczby zainteresowanych. Może właśnie Twój list przeważy szalę? Będziemy starali się reagować jak najszybciej, jednak droga od pomysłu do ukazania się na rynku zajmuje kilka miesięcy.

Jeśli uważasz, że wiesz coś ciekawego i chciałbyś podzielić się tym z pozostałymi Czytelnikami – napisz artykuł. Jak wskazuje nazwa naszej publikacji, pragniemy tworzyć wspólnie z Czytelnikami klan ludzi, którym zależy, aby mikrokomputery były wykorzystywane z pełną świadomością ich możliwości i ograniczeń. Możesz więc napisać o tym, co sam odkryłeś lub udoskonaliłeś, posługując się Spectrum, Commodorem, Amstradem czy IBM PC (jeśli zainspirował Cię pomysł już gdzieś publikowany, powołaj się na niego, załączając do tekstu wykaz literatury). Możesz napisać o możliwościach rozbudowy lub bardziej efektywnym sposobie programowania. Możemy też opublikować Twój program, jeśli będzie on interesujący dla innych i... niezbyt długi. To samo dotyczy konstrukcji sprzętowych. Ciekawe rozwiązania układowe, niestandardowe aplikacje, czyli forum do wymiany pomysłów i doświadczeń dla wszystkich, hobbistów i profesjonalistów.

Wymagania dotyczące przysyłanych materiałów:
tekst: w maszynopisie, z dwoma kopiami, (30 wierszy na stronie, 60 uderzeń w wierszu, czyli duży odstęp między liniami)
objętość: maksymalnie 7-9 stron maszynopisu (jasno i zwięźle),
rysunki: mogą być w ołówku (przejrzyste i czytelne)
wydruki: maksymalnie kontrastowe (załóż nową taśmę do drukarki, jeśli możesz przyślij program na nośniku magnetycznym – zwrócimy!)
programy: nie tylko przetestowane, ale też eleganckie
konstrukcje: wyłącznie uruchomione i koniecznie sprawdzone
ponadto: adres domowy, telefony – domowy i służbowy, ew. numer konta PKO.

W. Minkowski

W NASTĘPNYM ZESZYCIE KOMPUTER OSOBISTY „DIALOG"